

# TEORIA WSPÓŁBIEŻNOŚCI

## LABORATORIUM 7

Algorytm współbieżnej eliminacji Gaussa-Jordana

Bartosz Konieczny

Wydział Informatyki

Grupa 5

Wtorek 8:00

data oddania:

18.12.2025





## Spis treści

<b>1 Ćwiczenie .....</b>	<b>3</b>
<b>2 Przedstawienie problemu .....</b>	<b>4</b>
2.1 Opis algorytmu .....	4
2.2 Operacje potrzebne do realizacji algorytmu .....	4
2.3 Przykład .....	4
<b>3 Algorytm Gaussa-Jordana .....</b>	<b>9</b>
<b>4 Instrukcja uruchomienia .....</b>	<b>14</b>
<b>5 Format pliku wejściowego .....</b>	<b>14</b>
<b>6 Opis programu .....</b>	<b>14</b>
<b>7 Wynik działania dla przykładowych danych .....</b>	<b>15</b>
7.1 example1 .....	15
7.2 example2 .....	15
7.3 Inne przypadki .....	15



## Ćwiczenie

W części teoretycznej proszę:

1. zlokalizować niepodzielne czynności wykonywane przez algorytm, nazwać je oraz zbudować alfabet w sensie teorii śladów,
2. skonstruować relację zależności dla alfabetu, opisującego algorytm eliminacji Gaussa-Jordana,
3. przedstawić algorytm eliminacji Gaussa-Jordana w postaci ciągu symboli alfabetu,
4. wygenerować graf zależności Dekerta,
5. przekształcić ciąg symboli opisujący algorytm do postaci normalnej Foaty.

Proszę zaprojektować i zaimplementować współbieżny algorytm eliminacji Gaussa-Jordana. W szczególności proszę zwrócić uwagę na implementację jak najlepiej odwzorowującą graf zależności lub postać normalną Foaty.



## Przedstawienie problemu

### 2.1 Opis algorytmu

Celem algorytmu Gaussa jest doprowadzenie macierzy wejściowej ( $M$ ) do postaci macierzy trójkątnej, dzięki czemu łatwo obliczyć rozwiązanie układu równań. Algorytm Gaussa-Jordana ma za zadanie doprowadzić macierz  $M$  do postaci macierzy jednostkowej.

### 2.2 Operacje potrzebne do realizacji algorytmu

Do realizacji algorytmu musimy wykorzystać następujące niepodzielne zadania obliczeniowe:

- $A_i$  - znalezienie mnożnika dla elementu  $M_{i,i}$ , do normalizacji wiersza  $i$   
 $m_i = \frac{1}{M_{i,i}}$
- $B_{i,j}$  - pomnożenie  $j$ -tego elementu wiersza  $i$  przez mnożnik do normalizacji tego wiersza  
 $M_{i,j} = M_{i,j} \cdot m_i$
- $C_{i,k}$  - znalezienie mnożnika dla wiersza  $i$ , do odejmowania go od wiersza  $k$   
 $n_{i,k} = \frac{M_{k,i}}{M_{i,i}}$
- $D_{i,j,k}$  - pomnożenie  $j$ -tego elementu wiersza  $i$  przez mnożnik dla wiersza  $k$   
 $v_{i,j,k} = M_{i,j} \cdot n_{i,k}$
- $E_{i,j,k}$  - odjęcie  $j$ -tego elementu wiersza  $k$  od wiersza  $i$   
 $M_{i,j} = M_{i,j} - v_{k,j,i}$

### 2.3 Przykład

Aby lepiej zobrazować zasadę działania algorytmu rozpatrzmy przykład.

Dany jest układ równań (1)

$$\begin{cases} 2x_0 + 1x_1 + 3x_2 = 6 \\ 4x_0 + 3x_1 + 8x_2 = 15 \\ 6x_0 + 5x_1 + 16x_2 = 27 \end{cases} \quad (1)$$

Możemy zapisać go w postaci macierzowej (2)

$$\begin{bmatrix} 2 & 1 & 3 \\ 4 & 3 & 8 \\ 6 & 5 & 16 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 15 \\ 27 \end{bmatrix} \quad (2)$$

Dla uproszczenia jako (3)

$$\left[ \begin{array}{ccc|c} 2 & 1 & 3 & 6 \\ 4 & 3 & 8 & 15 \\ 6 & 5 & 16 & 27 \end{array} \right] \quad (3)$$

Normalizujemy zerowy wiersz, tak by jego zerowy element był równy 1. Dzieląc zerowy wiersz przez  $M_{0,0} = 2$  otrzymujemy (4)

Operacje:  $A_0, B_{0,0}, B_{0,1}, B_{0,2}, B_{0,3}$ .



$$\left[ \begin{array}{ccc|c} 1 & \frac{1}{2} & \frac{3}{2} & 3 \\ 4 & 3 & 8 & 15 \\ 6 & 5 & 16 & 27 \end{array} \right] \quad (4)$$

Następnie z wiersza 1 i 2 musimy wyeliminować elementy z zerowej kolumny. W tym celu odejmiemy od nich wiersz zerowy pomnożony odpowiednio przez 4 i 6 otrzymując (5)

Operacje:  $C_{0,1}, D_{0,0,1}, D_{0,1,1}, D_{0,2,1}, D_{0,3,1}, E_{1,0,0}, E_{1,1,0}, E_{1,2,0}, E_{1,3,0},$   
 $C_{0,2}, D_{0,0,2}, D_{0,1,2}, D_{0,2,2}, D_{0,3,2}, E_{2,0,0}, E_{2,1,0}, E_{2,2,0}, E_{2,3,0}.$

$$\left[ \begin{array}{ccc|c} 1 & \frac{1}{2} & \frac{3}{2} & 3 \\ 0 & 1 & 2 & 3 \\ 0 & 2 & 7 & 9 \end{array} \right] \quad (5)$$

Kolejny krok będzie identyczny jak pierwszy krok, jednak tym razem dla wiersza pierwszego. W związku z tym wykonujemy normalizację tego wiersza - dzielenie przez  $M_{1,1} = 1$ , w związku z tym nie będę prezentował tego kroku. Dalej eliminujemy  $M_{2,1}$  - poprzez odjęcie podwojonego wiersza pierwszego oraz  $M_{0,1}$  - poprzez odjęcie wiersza pierwszego pomnożonego razy  $\frac{1}{2}$ , jest to eliminacja pierwszej kolumny (6)

Operacje:  $A_1, B_{1,1}, B_{1,2}, B_{1,3},$   
 $C_{1,0}, D_{1,1,0}, D_{1,2,0}, D_{1,3,0}, E_{0,1,1}, E_{0,2,1}, E_{0,3,1},$   
 $C_{1,2}, D_{1,1,2}, D_{1,2,2}, D_{1,3,2}, E_{2,1,1}, E_{2,2,1}, E_{2,3,1}.$

$$\left[ \begin{array}{ccc|c} 1 & 0 & \frac{1}{2} & \frac{3}{2} \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 3 & 3 \end{array} \right] \quad (6)$$

Następnie normalizujemy wiersz 2 - element  $M_{2,2}$  musi być równy 1, więc dzielimy ten wiersz przez 3 otrzymując (7)

Operacje:  $A_2, B_{2,2}, B_{2,3}.$

$$\left[ \begin{array}{ccc|c} 1 & 0 & \frac{1}{2} & \frac{3}{2} \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 1 \end{array} \right] \quad (7)$$

Ostatnim już etapem będzie eliminacja drugiej kolumny - zerujemy elementy  $M_{1,2}$  oraz  $M_{0,2}$  poprzez odjęcie odpowiednio podwojonego wiersza 2 oraz wiersza 2 pomnożonego przez  $\frac{1}{2}$ .

Operacje:  $C_{2,0}, D_{2,2,0}, D_{2,3,0}, E_{0,2,2}, E_{0,3,2},$   
 $C_{2,1}, D_{2,2,1}, D_{2,3,1}, E_{1,2,2}, E_{1,3,2}.$

$$\left[ \begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{array} \right] \quad (8)$$

W związku z tym możemy wyznaczyć nasze rozwiązania układu równań (9)

$$\begin{cases} x_0 = 1 \\ x_1 = 1 \\ x_2 = 1 \end{cases} \quad (9)$$



W związku z tym nasz alfabet  $\Sigma$  to

$$\Sigma = \{$$

$$A_0, B_{0,0}, B_{0,1}, B_{0,2}, B_{0,3}, C_{0,1}, D_{0,0,1}, D_{0,1,1}, D_{0,2,1}, D_{0,3,1}, E_{1,0,0}, E_{1,1,0}, E_{1,2,0}, E_{1,3,0},$$

$$C_{0,2}, D_{0,0,2}, D_{0,1,2}, D_{0,2,2}, D_{0,3,2}, E_{2,0,0}, E_{2,1,0}, E_{2,2,0}, E_{2,3,0}, A_1, B_{1,1}, B_{1,2}, B_{1,3},$$

$$C_{1,0}, D_{1,1,0}, D_{1,2,0}, D_{1,3,0}, E_{0,1,1}, E_{0,2,1}, E_{0,3,1}, C_{1,2}, D_{1,1,2}, D_{1,2,2}, D_{1,3,2}, E_{2,1,1}, E_{2,2,1}, E_{2,3,1},$$

$$A_2, B_{2,2}, B_{2,3}, C_{2,0}, D_{2,2,0}, D_{2,3,0}, E_{0,3,2}, E_{0,2,2}, C_{2,1}, D_{2,2,1}, D_{2,3,1}, E_{1,3,2}, E_{1,2,2}$$

$$\}$$

Relacja zależności w tym przypadku zaprezentowana jest poniżej

$$D = \text{sym}\{ \{$$

$$(A_0, B_{0,0}), (A_0, B_{0,1}), (A_0, B_{0,2}), (A_0, B_{0,3}),$$

$$(B_{0,0}, C_{0,1}), (B_{0,0}, C_{0,2}),$$

$$(B_{0,1}, D_{0,1,1}), (B_{0,1}, D_{0,1,2}), (B_{0,2}, D_{0,2,1}), (B_{0,2}, D_{0,2,2}), (B_{0,3}, D_{0,3,1}), (B_{0,3}, D_{0,3,2}),$$

$$(C_{0,1}, D_{0,0,1}), (C_{0,1}, D_{0,1,1}), (C_{0,1}, D_{0,2,1}), (C_{0,1}, D_{0,3,1}),$$

$$(D_{0,0,1}, E_{1,0,0}), (D_{0,1,1}, E_{1,1,0}), (D_{0,2,1}, E_{1,2,0}), (D_{0,3,1}, E_{1,3,0}),$$

$$(C_{0,2}, D_{0,0,2}), (C_{0,2}, D_{0,1,2}), (C_{0,2}, D_{0,2,2}), (C_{0,2}, D_{0,3,2}),$$

$$(D_{0,0,2}, E_{2,0,0}), (D_{0,1,2}, E_{2,1,0}), (D_{0,2,2}, E_{2,2,0}), (D_{0,3,2}, E_{2,3,0}),$$

$$(A_1, B_{1,1}), (A_1, B_{1,2}), (A_1, B_{1,3}),$$

$$(E_{1,2,0}, B_{1,2}), (E_{1,3,0}, B_{1,3}),$$

$$(B_{1,1}, C_{1,0}), (B_{1,1}, C_{1,2}),$$

$$(E_{2,1,0}, C_{1,2}), (E_{2,2,0}, E_{2,2,1}), (E_{2,3,0}, E_{2,3,1}),$$

$$(C_{1,0}, D_{1,1,0}), (C_{1,0}, D_{1,2,0}), (C_{1,0}, D_{1,3,0}),$$

$$(B_{1,2}, D_{1,2,0}), (B_{1,3}, D_{1,3,0})$$

$$(D_{1,1,0}, E_{0,1,1}), (D_{1,2,0}, E_{0,2,1}), (D_{1,3,0}, E_{0,3,1}),$$

$$(C_{1,2}, D_{1,1,2}), (C_{1,2}, D_{1,2,2}), (C_{1,2}, D_{1,3,2}),$$

$$(B_{1,2}, D_{1,2,2}), (B_{1,3}, D_{1,3,2})$$

$$(D_{1,1,2}, E_{2,1,1}), (D_{1,2,2}, E_{2,2,1}), (D_{1,3,2}, E_{2,3,1}),$$

$$(A_2, B_{2,2}), (A_2, B_{2,3}),$$

$$(E_{2,3,1}, B_{2,3}),$$

$$(B_{2,2}, C_{2,0}), (B_{2,2}, C_{2,1}),$$

$$(B_{2,3}, D_{2,3,0}), (B_{2,3}, D_{2,3,1}),$$

$$(E_{0,2,1}, E_{0,2,2}), (E_{0,3,1}, E_{0,3,2}),$$

$$(C_{2,0}, D_{2,2,0}), (D_{2,2,0}, E_{0,2,2}), (C_{2,0}, D_{2,3,0}), (D_{2,3,0}, E_{0,3,2}),$$

$$(C_{2,1}, D_{2,2,1}), (D_{2,2,1}, E_{1,2,2}), (C_{2,1}, D_{2,3,1}), (D_{2,3,1}, E_{1,3,2})$$

$$\}^+ \} \cup I_\Sigma$$

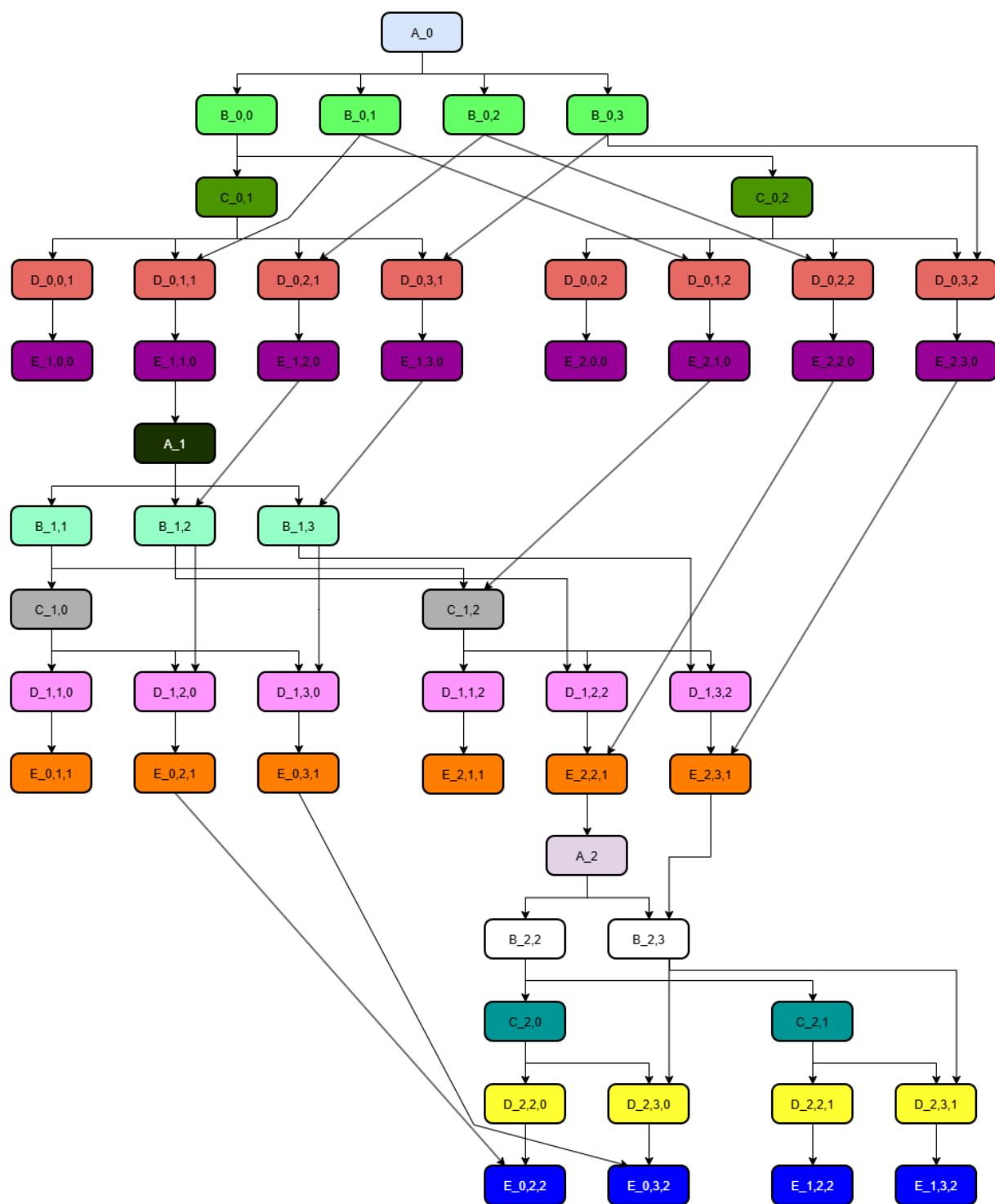


$$t = [w]_{\equiv_I^+} = [\langle A_0, B_{0,0}, B_{0,1}, B_{0,2}, B_{0,3}, C_{0,1}, D_{0,0,1}, D_{0,1,1}, D_{0,2,1}, D_{0,3,1}, E_{1,0,0}, E_{1,1,0}, E_{1,2,0}, E_{1,3,0}, \\ C_{0,2}, D_{0,0,2}, D_{0,1,2}, D_{0,2,2}, D_{0,3,2}, E_{2,0,0}, E_{2,1,0}, E_{2,2,0}, E_{2,3,0}, A_1, B_{1,1}, B_{1,2}, B_{1,3}, C_{1,0}, \\ D_{1,1,0}, D_{1,2,0}, D_{1,3,0}, E_{0,1,1}, E_{0,2,1}, E_{0,3,1}, C_{1,2}, D_{1,1,2}, D_{1,2,2}, D_{1,3,2}, E_{2,1,1}, E_{2,2,1}, E_{2,3,1}, \\ A_2, B_{2,2}, B_{2,3}, C_{2,0}, D_{2,2,0}, D_{2,3,0}, E_{0,3,2}, E_{0,2,2}, C_{2,1}, D_{2,2,1}, D_{2,3,1}, E_{1,3,2}, E_{1,2,2} \rangle]_{\equiv_I^+}$$

### Postać Normalna Foaty

$$t = [\langle A \rangle]_{\equiv_I^+} = [\{A_0\}]_{\equiv_I^+} \\ \neg [\{B_{0,0}, B_{0,1}, B_{0,2}, B_{0,3}\}]_{\equiv_I^+} \\ \neg [\{C_{0,1}, C_{0,2}\}]_{\equiv_I^+} \\ \neg [\{D_{0,0,1}, D_{0,1,1}, D_{0,2,1}, D_{0,3,1}, D_{0,0,2}, D_{0,1,2}, D_{0,2,2}, D_{0,3,2}\}]_{\equiv_I^+} \\ \neg [\{E_{1,0,0}, E_{1,1,0}, E_{1,2,0}, E_{1,3,0}, E_{2,0,0}, E_{2,1,0}, E_{2,2,0}, E_{2,3,0}\}]_{\equiv_I^+} \\ \neg [\{A_1\}]_{\equiv_I^+} \\ \neg [\{B_{1,1}, B_{1,2}, B_{1,3}\}]_{\equiv_I^+} \\ \neg [\{C_{1,0}, C_{1,2}\}]_{\equiv_I^+} \\ \neg [\{D_{1,1,0}, D_{1,2,0}, D_{1,3,0}, D_{1,1,2}, D_{1,2,2}, D_{1,3,2}\}]_{\equiv_I^+} \\ \neg [\{E_{0,1,1}, E_{0,2,1}, E_{0,3,1}, E_{2,1,1}, E_{2,2,1}, E_{2,3,1}\}]_{\equiv_I^+} \\ \neg [\{A_2\}]_{\equiv_I^+} \\ \neg [\{B_{2,2}, B_{2,3}\}]_{\equiv_I^+} \\ \neg [\{C_{2,0}, C_{2,1}\}]_{\equiv_I^+} \\ \neg [\{D_{2,2,0}, D_{2,3,0}, D_{2,2,1}, D_{2,3,1}\}]_{\equiv_I^+} \\ \neg [\{E_{0,2,2}, E_{0,3,2}, E_{1,2,2}, E_{1,3,2}\}]_{\equiv_I^+} = \\ [F_1]_{\equiv_I^+} \neg [F_2]_{\equiv_I^+} \neg [F_3]_{\equiv_I^+} \neg [F_4]_{\equiv_I^+} \neg [F_5]_{\equiv_I^+} \neg [F_6]_{\equiv_I^+} \neg [F_7]_{\equiv_I^+} \neg [F_8]_{\equiv_I^+} \neg [F_9]_{\equiv_I^+} \neg [F_{10}]_{\equiv_I^+} \\ \neg [F_{11}]_{\equiv_I^+} \neg [F_{12}]_{\equiv_I^+} \neg [F_{13}]_{\equiv_I^+} \neg [F_{14}]_{\equiv_I^+} \neg [F_{15}]_{\equiv_I^+}$$

Na podstawie powyższych wyprowadzeń można narysować i pokolorować graf Diekerta zaprezentowany na rysunku poniżej (Rys. 1).



Rys. 1: Graf Diekerta wraz z kolorowaniem klas foaty dla przykładu macierzy  $3 \times 3$ .





## Algorytm Gaussa-Jordana

Dane wejściowe w postaci kwadratowej macierzy  $M$  o wymiarach  $n \times n$  oraz wektora wynikowego  $y$  poniżej (10)

$$\begin{bmatrix} M_{0,0} & M_{0,1} & M_{0,2} & \cdots & M_{0,n-1} \\ M_{1,0} & M_{1,1} & M_{1,2} & \cdots & M_{1,n-1} \\ M_{2,0} & M_{2,1} & M_{2,2} & \cdots & M_{2,n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ M_{n-1,1} & M_{n-1,2} & M_{n-1,3} & \cdots & M_{n-1,n-1} \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{bmatrix} \quad (10)$$

Możemy przedstawić w postaci (11)

$$\left[ \begin{array}{ccccc|c} M_{0,0} & M_{0,1} & M_{0,2} & \cdots & M_{0,n-1} & y_0 \\ M_{1,0} & M_{1,1} & M_{1,2} & \cdots & M_{1,n-1} & y_1 \\ M_{2,0} & M_{2,1} & M_{2,2} & \cdots & M_{2,n-1} & y_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ M_{n-1,1} & M_{n-1,2} & M_{n-1,3} & \cdots & M_{n-1,n-1} & y_{n-1} \end{array} \right] \quad (11)$$

Normalizujemy wiersz 0 wykonując operacje:

- znalezienie mnożnika -  $A_0$
- pomnożenie elementów wiersza -  $B_{0,0}, B_{0,1}, B_{0,2}, \dots, B_{0,n}$

I otrzymujemy (12)

$$\left[ \begin{array}{ccccc|c} 1 & M_{(0,1)_0} & M_{(0,2)_0} & \cdots & M_{(0,n-1)_0} & y_{0_0} \\ M_{1,0} & M_{1,1} & M_{1,2} & \cdots & M_{1,n-1} & y_1 \\ M_{2,0} & M_{2,1} & M_{2,2} & \cdots & M_{2,n-1} & y_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ M_{n-1,1} & M_{n-1,2} & M_{n-1,3} & \cdots & M_{n-1,n-1} & y_{n-1} \end{array} \right] \quad (12)$$

Przystępujemy do wyzerowania kolumny 0. Wykonując operacje dla wiersza 1:

- znalezienie mnożnika dla wiersza 0 w przypadku wiersza 1 -  $C_{0,1}$
- pomnożenie elementów wiersza 0 w przypadku wiersza 1 -  $D_{0,0,1}, D_{0,1,1}, \dots, D_{0,n,1}$
- odjęcie elementów wiersza 0 od wiersza 1 -  $E_{1,0,0}, E_{1,1,0}, \dots, E_{1,n,0}$

Podobne operacje w przypadku wiersza 2:

- znalezienie mnożnika dla wiersza 0 w przypadku wiersza 2 -  $C_{0,2}$
- pomnożenie elementów wiersza 0 w przypadku wiersza 2 -  $D_{0,0,2}, D_{0,1,2}, \dots, D_{0,n,2}$
- odjęcie elementów wiersza 0 od wiersza 2 -  $E_{2,0,0}, E_{2,1,0}, \dots, E_{2,n,0}$

Oraz dla pozostałych wierszy, więc w tym kroku wykonamy operacje:

- $\forall_{i \in \{1,2,\dots,n-1\}} C_{0,i}$
- $\forall_{i \in \{0,1,\dots,n\}} \forall_{j \in \{1,2,\dots,n-1\}} D_{0,i,j}$
- $\forall_{i \in \{0,1,\dots,n\}} \forall_{j \in \{1,2,\dots,n-1\}} E_{j,i,0}$



W ten sposób otrzymamy (13)

$$\left[ \begin{array}{cccc|c} 1 & M_{(0,1)_0} & M_{(0,2)_0} & \cdots & M_{(0,n-1)_0} & y_{0_0} \\ 0 & M_{(1,1)_0} & M_{(1,2)_0} & \cdots & M_{(1,n-1)_0} & y_{1_0} \\ 0 & M_{(2,1)_0} & M_{(2,2)_0} & \cdots & M_{(2,n-1)_0} & y_{2_0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & M_{(n-1,2)_0} & M_{(n-1,3)_0} & \cdots & M_{(n-1,n-1)_0} & y_{(n-1)_0} \end{array} \right] \quad (13)$$

Normalizujemy wiersz 1 wykonując operacje:

- znalezieni mnożnika -  $A_1$
- pomnożenie elementów wiersza -  $B_{1,1}, B_{1,2}, \dots, B_{1,n}$

I otrzymujemy (14)

$$\left[ \begin{array}{cccc|c} 1 & M_{(0,1)_0} & M_{(0,2)_0} & \cdots & M_{(0,n-1)_0} & y_{0_0} \\ 0 & 1 & M_{(1,2)_1} & \cdots & M_{(1,n-1)_1} & y_{1_1} \\ 0 & M_{(2,1)_0} & M_{(2,2)_0} & \cdots & M_{(2,n-1)_0} & y_{2_0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & M_{(n-1,2)_0} & M_{(n-1,3)_0} & \cdots & M_{(n-1,n-1)_0} & y_{(n-1)_0} \end{array} \right] \quad (14)$$

Przystępujemy do wyzerowania kolumny 1, więc w tym kroku wykonamy operacje:

- $\forall_{i \in \{0\} \cup \{2, \dots, n-1\}} C_{1,i}$
- $\forall_{i \in \{1, \dots, n\}} \forall_{j \in \{0\} \cup \{2, \dots, n-1\}} D_{1,i,j}$
- $\forall_{i \in \{1, \dots, n\}} \forall_{j \in \{0\} \cup \{2, \dots, n-1\}} E_{j,i,1}$

W wyniku operacji otrzymamy (15)

$$\left[ \begin{array}{cccc|c} 1 & 0 & M_{(0,2)_1} & \cdots & M_{(0,n-1)_1} & y_{0_1} \\ 0 & 1 & M_{(1,2)_1} & \cdots & M_{(1,n-1)_1} & y_{1_1} \\ 0 & 0 & M_{(2,2)_1} & \cdots & M_{(2,n-1)_1} & y_{2_1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & M_{(n-1,3)_1} & \cdots & M_{(n-1,n-1)_1} & y_{(n-1)_1} \end{array} \right] \quad (15)$$

W ten sam sposób postępujemy dla kolejnych wierszy (od 2 do  $n-1$ ). Wykonamy operacje:

- $\forall_{k \in \{2, \dots, n-1\}} A_k$
- $\forall_{k \in \{2, \dots, n-1\}} \forall_{i \in \{k, \dots, n\}} B_{k,i}$
- $\forall_{k \in \{2, \dots, n-1\}} \forall_{i \in \{0, \dots, k-1\} \cup \{k+1, \dots, n-1\}} C_{k,i}$
- $\forall_{k \in \{2, \dots, n-1\}} \forall_{i \in \{k, \dots, n\}} \forall_{j \in \{0, \dots, k-1\} \cup \{k+1, \dots, n-1\}} D_{k,i,j}$
- $\forall_{k \in \{2, \dots, n-1\}} \forall_{i \in \{k, \dots, n\}} \forall_{j \in \{0, \dots, k-1\} \cup \{k+1, \dots, n-1\}} E_{j,i,k}$



Ostatecznie otrzymamy macierz jednostkową i wektor będący wynikiem działań naszego algorytmu (16)

$$\left[ \begin{array}{cccc|c} 1 & 0 & 0 & \cdots & 0 & y_{0_{n-1}} \\ 0 & 1 & 0 & \cdots & 0 & y_{1_{n-1}} \\ 0 & 0 & 1 & \cdots & 0 & y_{2_{n-1}} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & y_{(n-1)_{n-1}} \end{array} \right] \quad (16)$$

Alfabet

Dla uproszczenia zapisu skorzystam z następujących zbiorów:

- $A' = \{A_k : k \in \{0, \dots, n-1\}\},$
- $B' = \{B_{k,i} : k \in \{0, \dots, n-1\} \wedge i \in \{k, \dots, n\}\},$
- $C' = \{C_{k,i} : k \in \{0, \dots, n-1\} \wedge i \in \{0, \dots, k-1\} \cup \{k+1, \dots, n-1\}\},$
- $D' = \{D_{k,i,j} : k \in \{0, \dots, n-1\} \wedge i \in \{k, \dots, n\} \wedge j \in \{0, \dots, k-1\} \cup \{k+1, \dots, n-1\}\},$
- $E' = \{E_{j,i,k} : k \in \{0, \dots, n-1\} \wedge i \in \{k, \dots, n\} \wedge j \in \{0, \dots, k-1\} \cup \{k+1, \dots, n-1\}\}.$

Alfabet będzie miał postać (17)

$$\Sigma = A' \cup B' \cup C' \cup D' \cup E' \quad (17)$$

Do wyznaczania relacji zależności  $D$  wykorzystam zbiory:

- $A'' = \{(A_k, B_{k,i}) : k \in \{0, \dots, n-1\} \wedge i \in \{0, \dots, n\} \wedge k \leq i\}$
- $B'' = \{(B_{k,k}, C_{k,i}) : k, i \in \{0, \dots, n-1\} \wedge k \neq i\}$
- $B''' = \{(B_{k,i}, D_{k,i,j}) : k, j \in \{0, \dots, n-1\} \wedge i \in \{0, \dots, n\} \wedge k < i \wedge k \neq j\}$
- $C'' = \{(C_{k,i}, D_{k,j,i}) : k, i \in \{0, \dots, n-1\} \wedge j \in \{0, \dots, n\} \wedge k \neq i \wedge k \leq j\}$
- $D'' = \{(D_{k,j,i}, E_{i,j,k}) : k, j \in \{0, \dots, n-1\} \wedge j \in \{0, \dots, n\} \wedge k \neq i \wedge k \leq j\}$
- $E'' = \{(E_{k,i,k-1}, B_{k,i}) : k \in \{1, \dots, n-1\} \wedge i \in \{0, \dots, n\} \wedge k < i\}$
- $E''' = \{(E_{k,i,j}, C_{i,k}) : k, i, j \in \{0, \dots, n-1\} \wedge k \neq i \wedge k > j \wedge i > j\}$
- $E^{(4)} = \{(E_{k,i,j-1}, E_{k,i,j}) : k \in \{0, \dots, n-1\} \wedge i \in \{0, \dots, n\} \wedge j \in \{1, \dots, n-1\} \wedge i \geq k \wedge k \neq j-1 \wedge k \neq j\}$

Relacja zależności będzie miała postać (18)

$$D = \text{sym}\{\{A'' \cup B'' \cup B''' \cup C'' \cup D'' \cup E'' \cup E''' \cup E^{(4)}\}^+\} \cup I_\Sigma \quad (18)$$

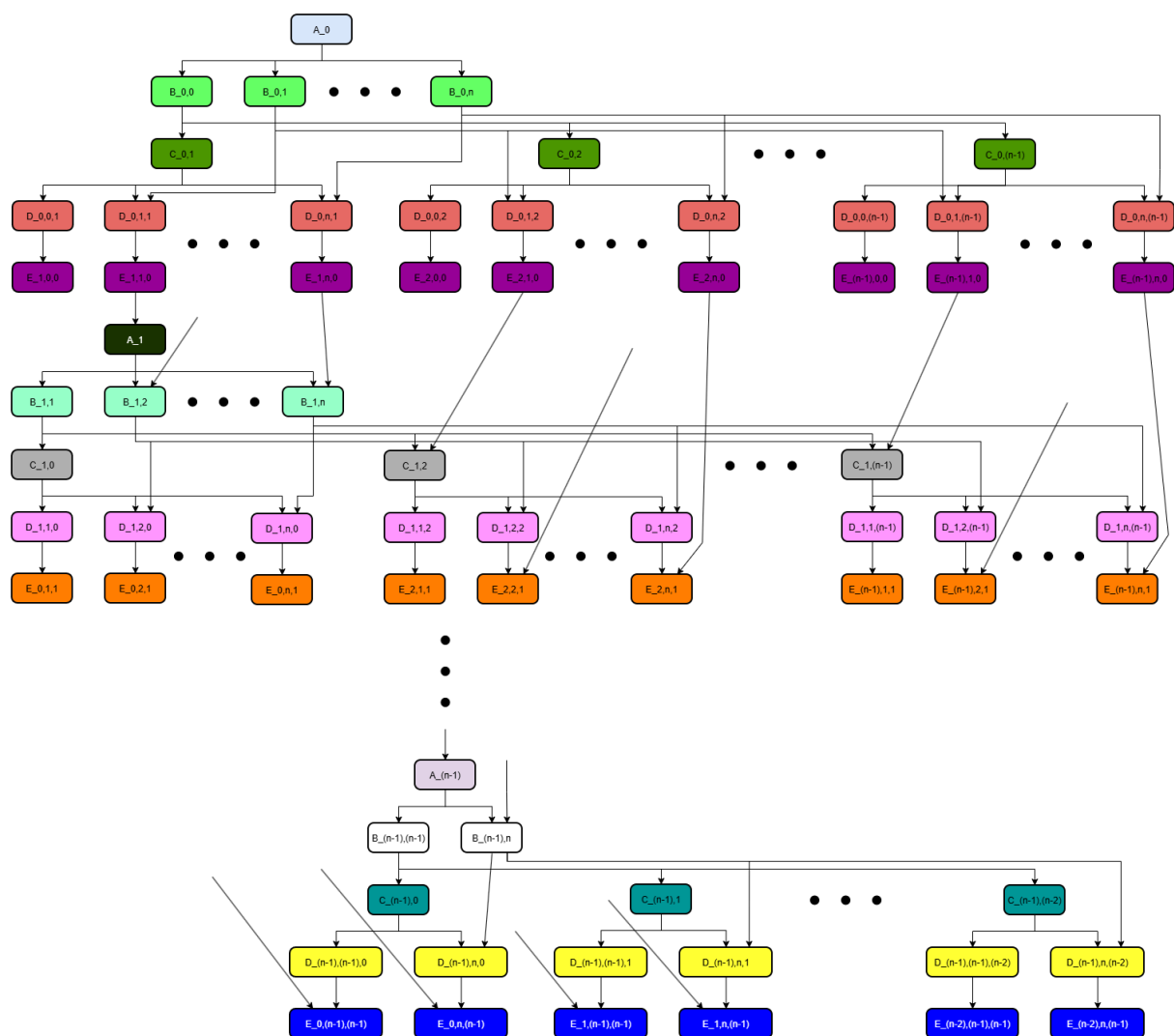
$$t = [w]_{\equiv_I^+} = \left[ \begin{array}{l} \langle A_0, B_{0,0}, \dots, B_{0,n}, C_{0,1}, \dots, C_{0,n-1}, D_{0,0,1}, \dots, D_{0,n,1}, D_{0,0,2}, \dots, D_{0,n,2}, \dots, D_{0,0,n-1}, \dots, D_{0,n,n-1}, \\ E_{1,0,0}, \dots, E_{1,n,0}, E_{2,0,0}, \dots, E_{2,n,0}, \dots, E_{n-1,0,0}, \dots, E_{n-1,n,0}, \\ \vdots, \\ A_{n-1}, B_{n-1,n-1}, B_{n-1,n}, C_{n-1,0}, \dots, C_{n-1,n-2}, D_{n-1,n-1,0}, D_{n-1,n,0}, \dots, D_{n-1,n-1,n-2}, D_{n-1,n,n-2}, \\ E_{0,n-1,n-1}, E_{0,n,n-1}, \dots, E_{n-2,n-1,n-1}, E_{n-2,n,n-1} \end{array} \right]_{\equiv_I^+}$$



## Postać Normalna Foaty

$$\begin{aligned}
 t = [\langle A \rangle]_{\equiv_I^+} &= [\{A_0\}]_{\equiv_I^+} \\
 &\frown [\{B_{0,0}, \dots, B_{0,n}\}]_{\equiv_I^+} \\
 &\frown [\{C_{0,1}, \dots, C_{0,n-1}\}]_{\equiv_I^+} \\
 &\frown [\{D_{0,0,1}, \dots, D_{0,n,1}, D_{0,0,2}, \dots, D_{0,n,2}, \dots, D_{0,0,n-1}, \dots, D_{0,n,n-1}\}]_{\equiv_I^+} \\
 &\frown [\{E_{1,0,0}, \dots, E_{1,n,0}, E_{2,0,0}, \dots, E_{2,n,0}, \dots, E_{n-1,0,0}, \dots, E_{n-1,n,0}\}]_{\equiv_I^+} \\
 &\vdots \\
 &\frown [\{A_k\}]_{\equiv_I^+} \\
 &\frown [\{B_{k,k}, \dots, B_{k,n}\}]_{\equiv_I^+} \\
 &\frown [\{C_{k,0}, \dots, C_{k,k-1}, C_{k,k+1}, \dots, C_{k,n-1}\}]_{\equiv_I^+} \\
 &\frown [\{D_{k,k,0}, \dots, D_{k,n,0}, \dots, D_{k,k,k-1}, \dots, D_{k,n,k-1}, D_{k,k,k+1}, \dots, D_{k,n,k+1}, \dots, D_{k,k,n-1}, \dots, D_{k,n,n-1}\}]_{\equiv_I^+} \\
 &\frown [\{E_{0,k,k}, \dots, E_{0,n,k}, \dots, E_{k-1,k,k}, \dots, E_{k-1,n,k}, E_{k+1,k,k}, \dots, E_{k+1,n,k}, \dots, E_{n-1,k,k}, \dots, E_{n-1,n,k}\}]_{\equiv_I^+} \\
 &\vdots \\
 &\frown [\{A_{n-1}\}]_{\equiv_I^+} \\
 &\frown [\{B_{n-1,n-1}, B_{n-1,n}\}]_{\equiv_I^+} \\
 &\frown [\{C_{n-1,0}, \dots, C_{n-1,n-2}\}]_{\equiv_I^+} \\
 &\frown [\{D_{n-1,n-1,0}, D_{n-1,n,0}, \dots, D_{n-1,n-1,n-2}, D_{n-1,n,n-2}\}]_{\equiv_I^+} \\
 &\frown [\{E_{0,n-1,n-1}, E_{0,n,n-1}, \dots, E_{n-2,n-1,n-1}, E_{n-2,n,n-1}\}]_{\equiv_I^+} = \\
 &[F_1]_{\equiv_I^+} \frown [F_2]_{\equiv_I^+} \frown [F_3]_{\equiv_I^+} \quad \dots \quad \frown [F_{5 \cdot n-1}]_{\equiv_I^+} \frown [F_{5 \cdot n}]_{\equiv_I^+}
 \end{aligned}$$

Na podstawie powyższych obliczeń prowadzących do wyznaczenia FNF można stworzyć graf Diekerta dla ogólnego przypadku macierzy (Rys. 2) oraz zaimplementować algorytm realizujący równoległe metodę Gaussa-Jordana.



Rys. 2: Graf Diekerta wraz z kolorowaniem klas foaty dla ogólnej macierzy  $n \times n$ .



## Instrukcja uruchomienia

Projekt został zaimplementowany w języku Java 25 i korzysta z systemu budowania Gradle. Do uruchomienia aplikacji wymagane jest środowisko JDK 25.

Aplikacja przyjmuje opcjonalny argument w postaci ścieżki do pliku wejściowego. Jeżeli argument nie zostanie podany, program domyślnie użyje pliku `input_example1.txt` z katalogu `src/main/resources/`, gdzie znajdują się również inne przykładowe pliki o różnych rozmiarach macierzy.

## Format pliku wejściowego

Program przyjmuje macierz kwadratową  $M$  o rozmiarze  $n \times n$  oraz wektor wynikowy  $y$ . Do aplikacji należy przekazywać pliki w formacie:

```
n
M_0,0 M_0,1 M_0,2 ... M_0,(n-1)
...
M_(n-1),0 M_(n-1),1 M_(n-1),2 ... M_(n-1),(n-1)
y_0 y_1 y_2 ... y_(n-1)
```

Plik wynikowy będzie znajdował się w tym samym katalogu, co plik wejściowy, a jego nazwa to `filename_out`, gdzie `filename` to nazwa pliku wejściowego.

Przykład poprawnego pliku wejściowego:

```
3
2.0 1.0 3.0
4.0 3.0 8.0
6.0 5.0 16.0
6.0 15.0 27.0
```

Program odczyta ten plik jako

$$\begin{bmatrix} 2 & 1 & 3 \\ 4 & 3 & 8 \\ 6 & 5 & 16 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 15 \\ 27 \end{bmatrix}$$

## Opis programu

Program został napisany w języku Java. Oraz został przetestowany przy pomocy [sprawdzarki](#) dostarczonej przez prowadzącego.

Program składa się z 4 głównych klas:

- `Main` - główna część programu odpowiedzialna za uruchomienie algorytmu dla zadanej macierzy,
- `FileManager` - część odpowiedzialna za parsowanie pliku wejściowego i utworzenie macierzy oraz zapisanie pliku wynikowego,
- `Matrix` - klasa reprezentująca macierz jako listę jednowymiarową dla szybszego działania,
- `GaussianElimination` - klasa implementująca algorytm Gaussa - Jordana.



## Wynik działania dla przykładowych danych

### 7.1 example1

Dla danego pliku `input_example1.txt`:

```
3
2.0 1.0 3.0
4.0 3.0 8.0
6.0 5.0 16.0
6.0 15.0 27.0
```

Otrzymamy plik `input_example1_out.txt`:

```
3
1.0 0.0 0.0
0.0 1.0 0.0
0.0 0.0 1.0
1.0 1.0 1.0
```

### 7.2 example2

Dla danego pliku `input_example2.txt` (plik dostępny w katalogu `resources`) zawierającego macierz o rozmiarze  $15 \times 15$ .

Otrzymamy plik `input_example2_out.txt` zawierający macierz jednostkową oraz wektor:

```
-0.5535598625142928 -0.05067651114069216 -0.3473165237239927 -1.6542570703056634
1.2551143962442728 -0.38145155231444716 -0.9539278207715004 -0.3477037633561322
-0.14874588242950004 0.6463610027084892 1.147144755703042 -0.06697242621465893
0.6037270621243773 0.8729251917017968 0.6301394487164557
```

### 7.3 Inne przypadki

W katalogu `resources` dołączone są również pliki zawierające pliki wejściowe i wyjściowe dla innych - większych przykładów macierzy.

Pozostałe przykładowe pliki:

- `input_example3.txt` - macierz  $20 \times 20$ ,
- `input_example4.txt` - macierz  $150 \times 150$ ,
- `input_example5.txt` - macierz  $300 \times 300$ ,
- `input_example6.txt` - macierz  $1000 \times 1000$ .

Wyniki dla wszystkich plików zostały przetestowane przez wspomnianą sprawdzarkę oraz pozytywnie przeszły jej testy.