Eliminacja Gaussa

Paweł Gorgolewski

20 grudnia 2022

1 Niepodzielne zadania obliczeniowe

$M_{1,1}$	$M_{1,2}$	$M_{1,3}$	$M_{1,4}$	$M_{1,5}$
$M_{2,1}$	$M_{2,2}$	$M_{2,3}$	$M_{2,4}$	$M_{2,5}$
$M_{3,1}$	$M_{3,2}$	$M_{3,3}$	$M_{3,4}$	$M_{3,5}$
$M_{4,1}$	$M_{4,2}$	$M_{4,3}$	$M_{4,4}$	$M_{4,5}$

Tabela 1: Pogladowa macierz (przykład 4x4, gdzie kolumna 5 to dodany wektor y)

Powyższy przykład jest aby zilustrować sytuację. Przy rozważaniach rozmiar macierzy nie jest zdefiniowany i oznaczamy go jako n.

Lista operacji na macierzy:

- $A_{r,i}$ - obliczenie mnożnika do zerowania kolumny i w wierszu r ($m_{r,i} = \frac{M_{r,i}}{M_{i,i}}$)
- $B_{r,j,i}$ - pomnożenie j-tego elementu wiersza i przez mnożnik - obliczenie odjemnej dla j-tego elementu w wierszu r ($n_{r,j,i} = M_{i,j} \cdot m_{r,i}$)
- $C_{r,j,i}$ - odjęcie odjemnej od j-tego elementu wiersza r ($M_{r,j} = M_{r,j} - n_{r,j,i}$)

2 Opis eliminacji Gaussa

Algorytm polega na kolejnym zerowaniu odpowednich elementów macierzy. Aby zdefiniować cały ciąg operacji eliminacji Gaussa na macierzy o rozmiarze n, zdefinijmy ciąg $z_{r,i}$, który zeruje element $M_{r,i}$:

$$z_{r,i} = A_{r,i}, B_{r,i,i}, C_{r,i,i}, B_{r,i+1,i}, C_{r,i+1,i}, ..., B_{r,n+1,i}, C_{r,n+1,i}$$

Zdefiniujmy jeszcze ciąg c_i, który odpowiada wyzerowaniu kolumny i (od elementu $M_{i+1,i}$ do $M_{n,i}$):

$$c_i = z_{i+1,i}, z_{i+2,i}, ..., z_{n,i}$$

Korzystając z definicji ciągu c, bardzo łatwo zdefiniować ciąg g, opisujący kolejne operacje w eliminacji Gaussa:

$$g = c_1, c_2, c_3, ..., c_{n-2}, c_{n-1}$$

3 Alfabet w sensie teorii śladów

$$\Sigma_A = \{A_{r,i} | 1 < r \leq n \land 1 \leq i < r\}$$

$$\Sigma_B = \{B_{r,j,i} | 1 < r \leq n \land 1 \leq i < r \land i \leq j \leq n+1\}$$

$$\Sigma_C = \{C_{r,j,i} | 1 < r \leq n \land 1 \leq i < r \land i \leq j \leq n+1\}$$

$$\Sigma = \Sigma_A \cup \Sigma_B \cup \Sigma_C$$

4 Relacje zależności

Dla każdej linijki, najpierw szukamy mnożnika, później wykonujemy mnożenie.

$$D_1 = \{(A_{r,i}, B_{r,j,i}) | A_{r,i} \subseteq \Sigma \wedge B_{r,j,i} \subseteq \Sigma\}$$

Dla każdej linijki, najpierw wykonujemy mnożenie, później odejmujemy.

$$D_2 = \{(B_{r,j,i}, C_{r,j,i}) | B_{r,j,i} \subseteq \Sigma \wedge C_{r,j,i} \subseteq \Sigma\}$$

Najpierw wykonujemy odejmowanie, później szukamy mnożnika. Skoro w operacji $A_{r,i}$ tworzymy mnożnik jako $m_{r,i} = \frac{M_{r,i}}{M_{i,i}}$, to jest on zależny od wcześniejszych operacji odejmowania (zerowanie poprzedniej kolumny) w wierszach r oraz i dla kolumny i.

$$D_3 = \{(C_{rc,jc,ic}, A_{ra,ia}) | C_{rc,jc,ic} \subseteq \Sigma \wedge A_{ra,ia} \subseteq \Sigma \wedge jc = ia \wedge (rc = ra \vee rc = ia) \wedge ia = ic + 1\}$$

Najpierw musimy mieć odjęte linijki, później możemy wykonać mnożenie. W operacji $B_{r,j,i}$ wykonujemy mnożenie, wykorzystując element $M_{i,j}$ z macierzy. Aby to wykonać musimy najpierw zakończończyć operację odejmowania $C_{i,j,i-1}$ (zerowanie wcześniejszej kolumny)

$$D_4 = \{(C_{rc,j,ic}, B_{rb,j,ib}) | C_{rc,j,ic} \subseteq \Sigma \wedge B_{rb,j,ib} \subseteq \Sigma \wedge rc = ib \wedge ib = ic + 1\}$$

Najpierw zerujemy kolumnę i, a później *i+1*.

$$D_5 = \{(C_{r,j,i1}, C_{r,j,i2}) | C_{r,j,i1} \subseteq \Sigma \wedge C_{r,j,i2} \subseteq \Sigma \wedge i2 = i1 + 1\}$$

Ostatecznie, otrzymujemy relację zależności jako:

$$D = sym\{\{D_1 \cup D_2 \cup D_3 \cup D_4 \cup D_5\}^+\} \cup I_\Sigma$$

Relacja niezależności to:

$$I = \Sigma^2 - D$$

5 Klasy Foaty

Mimo, iż rozważana macierz może być bardzo duża, zdefiniowanie klas Foaty nie jest problematyczne. Zdefiniujmy następujące zbiory (tak jak wyżej, oznaczamy n jako rozmiar macierzy):

$$F_{A,i} = \{A_{r,i} | i < r \leq n\}$$
$$F_{B,i} = \{B_{r,j,i} | i < r \leq n \wedge 1 < j \leq n + 1\}$$
$$F_{C,i} = \{C_{r,j,i} | i < r \leq n \wedge 1 < j \leq n + 1\}$$

Uporządkowane klasy Floaty:

$$(F_{A,1})(F_{B,1})(F_{C,1})(F_{A,2})(F_{B,2})(F_{C,2})...(F_{A,n-1})(F_{B,n-1})(F_{C,n-1})$$

6 Graf Diekerta

Aby stworzyć graf Diekerta, musimy znaleźć wszystkie bezpośrednie zależności pomiędzy operacjami (będą to krawędzie). Korzystając z rozważań z punktu 4, możemy stworzyć podobne podzbiory, lecz tym razem opisujące tylko bezpośrednie zależnośći.

Pierwsze trzy zbiory są identyczne z odpowiednio D_1, D_2 i D_3.

$$S_1 = \{(A_{r,i}, B_{r,j,i}) | A_{r,i} \subseteq \Sigma \wedge B_{r,j,i} \subseteq \Sigma\}$$

$$S_2 = \{(B_{r,j,i}, C_{r,j,i}) | B_{r,j,i} \subseteq \Sigma \wedge C_{r,j,i} \subseteq \Sigma\}$$

$$S_3 = \{(C_{rc,jc,ic}, A_{ra,ia}) | C_{rc,jc,ic} \subseteq \Sigma \wedge A_{ra,ia} \subseteq \Sigma \wedge jc = ia \wedge (rc = ra \vee rc = ia) \wedge ia = ic + 1\}$$

S_4 nie może być identyczne jak D_4, ponieważ powstaną krawędzie niebezpośrednie. Niektóre z odejmowań C mają bezpośredni wpływ na tworzenie mnożnika poprzez operację A (w S_3 istnieją zależności C-A, zaś w S_1 A-B). Musimy je wykluczyć.

$$S_4 = \{(C_{rc,j,ic}, B_{rb,j,ib}) | C_{rc,j,ic} \subseteq \Sigma \wedge B_{rb,j,ib} \subseteq \Sigma \wedge rc = ib \wedge ib = ic + 1 \wedge j \neq ib\}$$

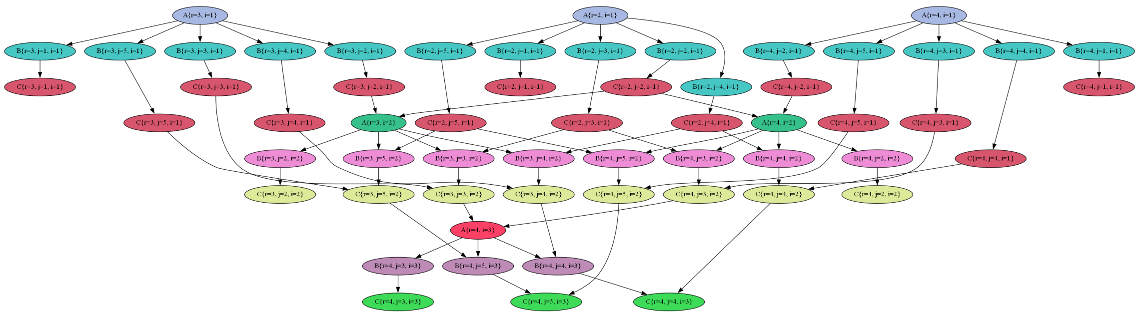
Z S_5 jest identycznie - odejmowania mają wpływ na tworzenie mnożnika, a on na mnożenie i kolejne odejmowanie.

$$S_5 = \{(C_{r,j,i1}, C_{r,j,i2}) | C_{r,j,i1} \subseteq \Sigma \wedge C_{r,j,i2} \subseteq \Sigma \wedge i2 = i1 + 1 \wedge j \neq i2\}$$

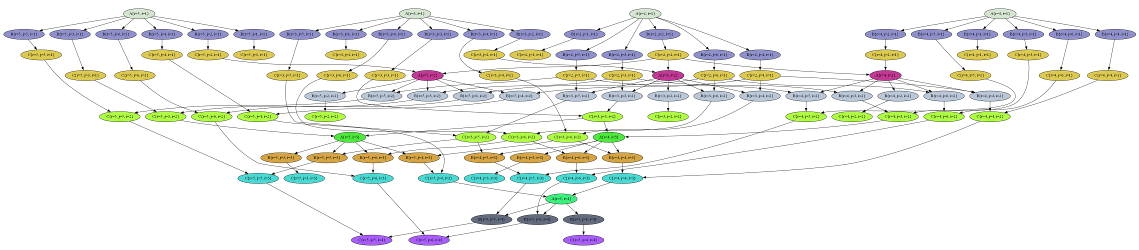
Zbiór wszystkich bezpośrednich zależności:

$$S = S_1 \cup S_2 \cup S_3 \cup S_4 \cup S_5$$

Poniżej dwa przykłady grafów Diekerta. Pierwszy dla macierzy 4x4, drugi 5x5.

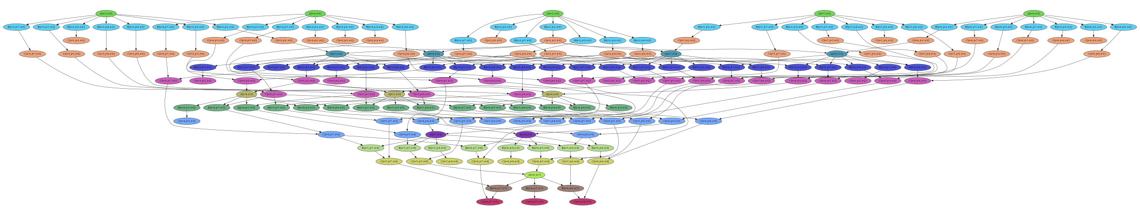


Rysunek 1: Graf Diekerta dla macierzy 4x4



Rysunek 2: Graf Diekerta dla macierzy 5x5

Ciekawość nie zna granic, dlatego poniżej umieszony został graf Diekerta dla macierzy 6x6 (dla celów poglądowych).



Rysunek 3: Graf Diekerta dla macierzy 6x6

7 Implementacja

Do implementacj wykorzystana została Java 19, Maven oraz Graphviz. Program można odpalić w następujący sposób:

```
mvn compile exec:java -D exec.mainClass=pl.agh.edu.tw.Main
-D exec.args=<your_input_file>
```

Program oprócz wygenerowania pliku *out.txt*, tworzy również graf Diekerta w postaci dot (*graph.dot*) oraz w postaci obrazu (*graph.png*)