Eliminacja Gaussa

Jakub Koźlak

Grudzień 2021

1 Wstęp

Zadanie polegało na implementacji eliminacji Gaussa z użyciem Teorii śladów.

$$\begin{bmatrix} M_{0,0} & M_{0,1} & M_{1,2} & \cdots & M_{0,n} \\ M_{1,0} & M_{1,1} & M_{1,2} & \cdots & M_{1,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ M_{n-1,0} & M_{n-1,1} & M_{n-1,2} & \cdots & M_{n-1,n} \end{bmatrix}$$

Niepodzielne operacje
Definiujemy niepodzielne operacje wykonywane na macierzy. Przyjąłem inne oznaczenia niż na zajęciach: i oznacza element przekątnej, j kolumnę, a k aktualnie redukowany wiersz macierzy M.

$A_{i,k}$ - znajdowanie mnożnika dla wiersza i, do odejmowania od wiersza k, $m_{k,i} = M_{k,i}/M_{i,i}$

$B_{i,k,j}$ - pomnożenie j-tego elementu wiersza przez $m_{i,k}$, utworzenie odjemnika $n_{i,k,j} = M_{i,j} * m_{i,k}$

$C_{i,k,j}$ - odjęcie j-tego elementu wiersza i od wiersza k, $M_{k,j} = M_{k,j} - n_{k,j,i}$

Alfabet: Teoria Śladów

$$\Sigma = \{A_{i,k} \mid 0 \leqslant i < n, i < k < n\} \cup$$

$$\{B_{i,k,j} \mid 0 \leqslant i < n,\ i < k \leqslant n,\ 0 \leqslant j \leqslant n\} \cup$$

$$\{C_{i,k,j} \mid 0 \leqslant i < n,\ i < k \leqslant n,\ 0 \leqslant j \leqslant n\}$$

Indeksy zostały przyjęte tak jak w implementacji: od 0 do N, dla macierzy rozszerzonej $M[N][N+1]$

Relacja Zależności

$$D = sym((D_1 \cup D_2 \cup D_3 \cup D_4 \cup D_5)^+) \ \cup \ I$$

$D_1 = \{(A_{i,k}, B_{i,k,j}) \mid A_{i,k}, B_{i,k,j} \ \in \ \Sigma\}$
$D_2 = \{(B_{i,k,j}, C_{i,k,j}) \mid B_{i,k,j}, C_{i,k,j} \ \in \ \Sigma\}$
$D_3 = \{(B_{i,k,j}, A_{x,k}) \mid B_{i,k,j}, A_{i,k} \ \in \ \Sigma \ gdzie \ x > i\}$
$D_4 = \{(C_{x,k,j}, A_{a,k}) \mid C_{x,k,j}, A_{a,k} \ \in \ \Sigma \ gdzie \ a = x\}$
$D_5 = \{(C_{i,k,j}, C_{x,k,j}) \mid C_{i,k,j}, C_{i,k,j} \ \in \ \Sigma \ gdzie \ x > i\}$

Relacja Niezależności

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

Zbiór relacji niezależności wyznaczamy poprzez różnicę alfabetu zbioru relacji zależności.

Algorytm i postać normalna Foaty

Pomimo skomplikowanego zapisu, klasy Foaty to:

$$[F_{A1}][F_{B1}][F_{C1}] \ ... \ [F_{AN}][F_{BN}][F_{CN}]$$

$$F_A = \{A_{i,k} \ |0 \leqslant i < n, i < k < n\}$$

$$F_B = \{B_{i,k,j} \mid 0 \leqslant i < n, \ i < k < n, \ 0 \leqslant j \leqslant \ n\}$$

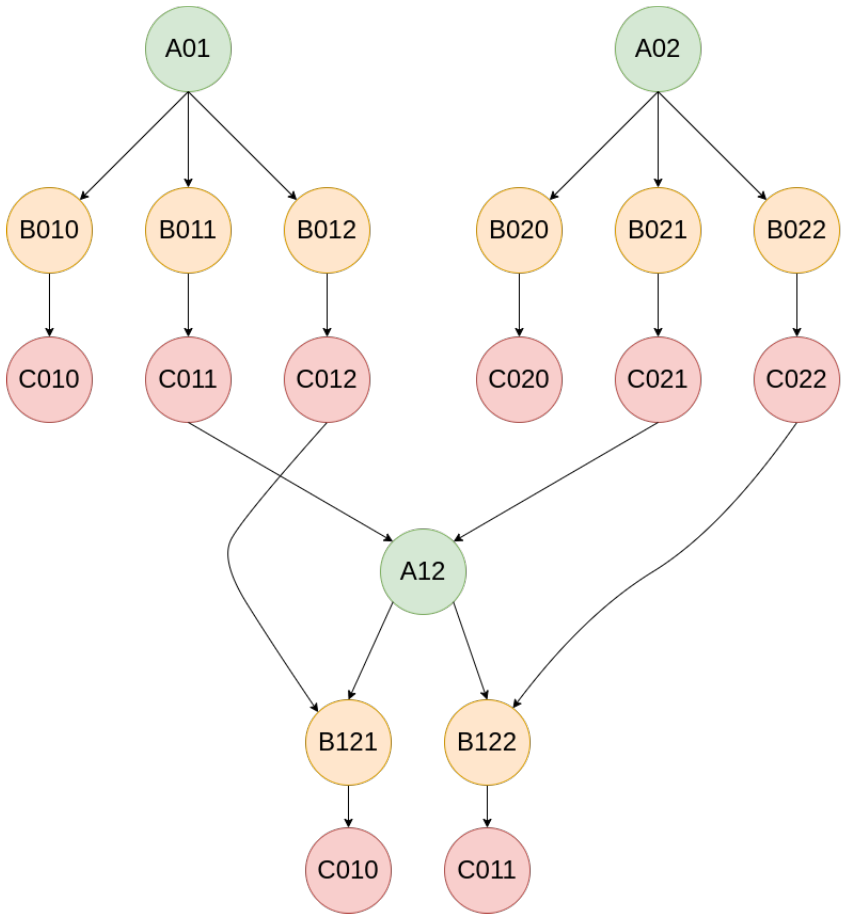
$$F_C = \{C_{i,k,j} \mid 0 \leqslant i < n, \ i < k < n, \ 0 \leqslant j \leqslant \ n\}$$

Eliminację Gaussa możemy przedstawić za pomocą wcześniej wspomnianych operacji. Polega ona na znalezieniu mnożnika ($A_{i,k}$), pomnożeniu odpowiednich elementów ($B_{i,k,j}$) oraz ich odjęciu ($C_{i,k,j}$).

$$A_{i,k} \rightarrow B_{i,k,i..n-1} \rightarrow C_{i,k,i..n}$$

Powtarzamy dla kolejnych elementów.

Graf Diekerta



Rysunek 1: Graf dla macierzy 3X3

Implementacja

- Powyższy problem rozwiązałem w języku Java.
- Program wczytuje macierze z pliku tekstowego znajdujących się w głownym katalogu.
- Klasa Scheduler: wyznacza FNF oraz przeprowadza obliczenia. Możliwość śledzenia jej działania ($isSilent = true$;).
- Poszczególne niepodzielne operacje: klasy A,B,C w pakiecie $operations$.
- $matrixPrepareService$: paczka zawierająca kod służący do przygotowania macierzy do obliczeń.