Teoria Współbieżności

Zadanie domowe – CW6

Joanna Klimek

15.12.2021 r.

Zadanie polega na wykonaniu następujących etapów (dla macierzy o rozmiarze *N*):

1. Zlokalizować niepodzielne czynności wykonywane przez algorytm i nazwać je.

Macierz uzupełniona, utworzona poprzez dołączenie wektora wyrazów wolnych, dla wymiarów 3x3:

$$\left[\begin{array}{ccc|c} M_{1,1} & M_{1,2} & M_{1,3} & M_{1,4} \\ M_{2,1} & M_{2,2} & M_{2,3} & M_{2,4} \\ M_{3,1} & M_{3,2} & M_{3,3} & M_{3,4} \end{array}\right]$$

Naszym celem jest otrzymanie macierzy w postaci trójkątnej górnej. Aby to uzyskać powtarzamy zadanie zerowania elementów kolumny numer j poniżej elementu w wierszu numer i.

Przykładowo:

$$\begin{bmatrix} \ddots & & \\ & M_{i,j} & \cdots \\ & \vdots & \ddots \\ & M_{k,j} & \end{bmatrix}$$

Aby wyzerować element $M_{k,j}$ musimy odjąć od niego element $M_{i,j}$ przemnożony przez odpowiedni mnożnik.

Wyróżniamy 3 typy zadań:

- $(A_{k,i})$ znalezienie mnożnika: $m_{k,i} = M_{k,i}/M_{i,i}$
- $(B_{i,j,k})$ mnożenie: $c_{i,j,k} = M_{i,j} * m_{k,i}$
- $(C_{i,j,k})$ odejmowanie: $M_{k,j} = M_{k,j} - c_{i,j,k}$

gdzie

i – numer wiersza, wykorzystywanego do odejmowania
k – numer wiersza, od którego odejmujemy, chcąc wyzerować element
j – numer kolumny, w której chcemy wyzerować elementy poniżej wiersza i

2. Zbudować alfabet w sensie teorii śladów.

Dla macierzy o wymiarach 3×3 (nie uzupełnionej):

$$\Sigma = \{A_{k,i}, B_{i,j,k}, C_{i,j,k} \colon \begin{array}{l} i = 1 \dots 2, \\ k = 2 \dots 3, \\ j = i \dots 4\} \end{array}$$

Co możemy uogólnić dla macierzy o wymiarach $n \times n$ (nie uzupełnionej):

$$\Sigma = \{A_{k,i}, B_{i,j,k}, C_{i,j,k} \colon \begin{array}{l} i = 1, \dots n - 1, \\ k = i + 1, \dots n, \\ j = i, \dots n + 1\} \end{array}$$

3. **Skonstruować relacje (nie) zależności dla alfabetu, opisującego algorytm eliminacji Gaussa.**

W tym celu szukam zmiennych zapisywanych w jednej operacji, które są jednocześnie wykorzystywane w drugich.

Zauważam, że mnożenie elementu jest zależne od znalezionego odpowiadającego mnożnika (przy zgodnych i oraz k), czyli:

$$D_1 = \{(A_{k,i}, B_{i,j,k})\}$$

Zależne zmienne oznaczone w równaniach:

$(A_{k,i})\quad m_{k,i} = M_{k,i}/M_{i,i}$
$(B_{i,j,k})\quad c_{i,j,k} = M_{i,j} * m_{k,i}$

Podobnie odejmowanie jest zależne od mnożenia dla tych samych i, j, k:

$$D_2 = \{(B_{i,j,k}, C_{i,j,k})\}$$

Zależne zmienne oznaczone w równaniach:

$(B_{i,j,k})\quad c_{i,j,k} = M_{i,j} * m_{k,i}$
$(C_{i,j,k})\quad M_{k,j} = M_{k,j} - c_{i,j,k}$

Odejmowanie dla danych i, j, k modyfikuje $M_{k,j}$, które następnie może być wykorzystywane w mnożeniu pod warunkiem, że $i_{mnożenia} = k_{odejmowania}$, co można zapisać formalnie:

$$D_3 = \{(C_{ic,j,kc}, B_{ib,j,kb}) : i_b = k_c\}$$

Zależne zmienne oznaczone w równaniach:

$(B_{ib,j,kb})\quad c_{ib,j,kb} = M_{ib,j} * m_{kb,ib}$
$(C_{ic,j,kc})\quad M_{kc,j} = M_{kc,j} - c_{ic,j,kc}$

Odejmowanie dla danych i, j, k modyfikuje $M_{k,j}$, które następnie może być wykorzystywane w znajdywaniu mnożnika pod warunkami, które można zapisać formalnie:

$$D_4 = \{(A_{ka,ia}, C_{ic,jc,kc}) : (i_a = j_c \wedge k_a = k_c) \vee (i_a = k_c \wedge i_a = j_c)\}$$

Zależne zmienne oznaczone w równaniach:

$(A_{ka,ia})\quad m_{ka,ia} = M_{ka,ia}/M_{ia,ia}$
$(C_{ic,jc,kc})\quad M_{kc,jc} = M_{kc,jc} - c_{ic,jc,kc}$

Operacje odejmowania dla różnych i są również zależne:

$$D_5 = \{(C_{i1,j,k}, C_{i2,j,k})\}$$

Zależne zmienne oznaczone w równaniach:

$(C_{i1,j,k})$ $M_{k,j} = M_{k,j} - c_{i1,j,k}$

$(C_{i2,j,k})$ $M_{k,j} = M_{k,j} - c_{i2,j,k}$

Dla wszystkich zbiorów powyżej zachodzi oczywisty warunek: $A_{k,i}, B_{i,j,k}, C_{i,j,k} \epsilon \Sigma$

Podsumowując relacja zależności z uwzględnieniem przechodniości i symetrii może być zapisana jako:

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

Natomiast relacja niezależności:

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

4. **Przedstawić algorytm eliminacji Gaussa w postaci ciągu symboli alfabetu.**

Przyjmuję oznaczenie:

$p_{k,i}$ – wyzerowanie elementu $M_{k,j}$, poprzez odjęcie od wiersza k-tego wiersza i-tego, przemnożonego przez odpowiedni mnożnik

Wizualizacja:

$$\begin{bmatrix} \ddots & & & \\ & M_{i,j} & \cdots & \\ & \vdots & \ddots & \\ & M_{k,j} & & \end{bmatrix}$$

$p_{k,i}$ można rozpisać jako:

$$A_{k,i}, B_{i,i,k}, C_{i,i,k}, B_{i,i+1,k}, C_{i,i+1,k}, \ldots\ B_{i,n,k}, C_{i,n,k}$$

gdzie $A_{k,i}$ oznacza znalezienie mnożnika, a powtarzające się pary operacji $B_{i,j,k}, C_{i,j,k}$ przekształcają kolejne elementy w k-tym wierszu.

W takim wypadku eliminację Gaussa można zapisać w uproszczeniu jako:

$$(p_{2,1}, p_{3,1}, \ldots\ p_{n,1}, p_{3,2}, p_{4,2}, \ldots\ p_{n,2}, \ldots \ldots, p_{n-1,n-2}, p_{n,n-2}, p_{n,n-1})$$

gdzie przykładowo fragment $(p_{2,1}, p_{3,1}, \ldots\ p_{n,1})$ zeruje pierwszą kolumnę, a $(p_{n,n-1})$ ostatni element w drugiej od końca kolumnie.

5. Wygenerować graf zależności Diekerta.

Zbiór krawędzi grafu odpowiada relacjom zależności z pewnymi ograniczeniami na indeksach w celu zminimalizowania grafu i pozostawienia jedynie bezpośrednich zależności:

$$E_1 = \{(A_{k,i}, B_{i,j,k})\}$$

$$E_2 = \{(B_{i,j,k}, C_{i,j,k})\}$$

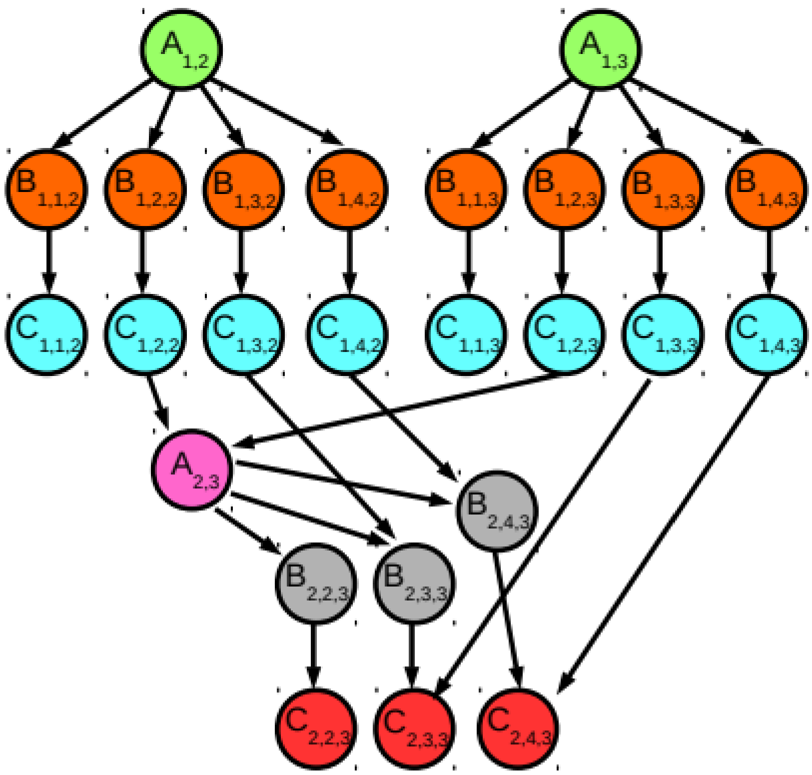
$$E_3 = \{(C_{ic,j,kc}, B_{ib,j,kb}) : i_b = k_c \ \land \ i_c = i_b - 1 \ \land j \neq i_b\}$$

$$E_4 = \{(C_{i1,j,k}, C_{i2,j,k}) : \ i_1 = i_2 - 1 \ \land j \neq i_2\}$$

$$E_5 = \{(C_{ic,jc,kc}, A_{ka,ia}) : (i_a = j_c \land k_a = k_c \ \land \ i_c = i_a - 1) \lor (i_a = k_c \land i_a = j_c \ \land \ i_c = i_a - 1)\}$$

$$E = E_1 \cup E_2 \cup E_3 \cup E_4 \cup E_5$$

Przykładowy graf analizowany na zajęciach:



6. Przekształcić ciąg symboli opisujący algorytm do postaci normalnej Foaty.

Klasy Foaty można podzielić na 3 typy:

- operacje znalezienia wszystkich mnożników potrzebnych do wyzerowania danej kolumny (F_{Am})
- operacje przemnażania przez znalezione mnożniki elementu na przekątnej w zerowanej kolumnie (F_{Bm})
- operacje odejmowania od kolejnych elementów w kolumnie wyliczonych współczynników (F_{Cm})

Przyjmując $m = 1,2, \dots n-1$ tworzę zbiory operacji następująco:

$$F_{Am} = \{A_{k,m}: \quad k = m+1, \dots n\}$$
$$F_{Bm} = \{B_{m,j,k}: \quad k = m+1, \dots n, \qquad j = m, \dots n+1\}$$
$$F_{Cm} = \{C_{m,j,k}: \quad k = m+1, \dots n, \qquad j = m, \dots n+1\}$$

Wtedy postać normalna Foaty:

$$[F_{A1}][F_{B1}][F_{C1}][F_{A2}][F_{B2}][F_{C2}] \dots \left[F_{A(n-1)}\right]\left[F_{B(n-1)}\right][F_{C(n-1)}]$$

Fragment $[F_{Am}][F_{Bm}][F_{Cm}]$ interpretujemy jako zerowanie kolumny o numerze n.

7. Proszę zaprojektować i zaimplementować równoległy algorytm eliminacji Gaussa. W szczególności proszę zwrócić uwagę na implementację jak najlepiej odwzorowującą graf zależności.

Implementacja znajduje się w załączonym archiwum.