Pralelni računarski sistemi

Vektorizacija gnezda petlji

Analiza zavisnosti kod ugnježđenih petlji i višedimenzionalnih polja

- Kod vektorizacije gnezda petlji vrši se vektorizacija najdublje ugnježđene petlje.
  - Zbog činjenice da sada pristup elementu polja zavisi od više iterativnih indeksa analiza zavisnosti po podacima je sada mnogo kompleksnija.
- Primer: dve ugnježđene petlje:

```
for i:=1 to x do
for j := 1 to y do
s1: A[a*i+b*j+c,d*i+e*j+f] = · · ·
s2 : · · · = A[g*i'+h*j'+k,l*i'+m*j'+n]
```

- Zavisnost izmedju naredbi s1 i s2 postoji ako postoje dva iterativna vektora $\alpha=(i, j)$ i $\beta=(i', j')$, pri čemu je $\alpha < \beta$, takva da važi

$$a * i + b * j + c = g * i' + h * j' + k$$
$$d * i + e * j + f = l * i' + m * j' + n$$

- Potrebno je rešiti sistem Diofantovih jednačina
  - np kompleksan problem

Opšti problem

```
DO i1 = L1, U1
  DO i2 = L2, U2
                ...
        DO in = Ln, Un
S1                A(f1(i1,...,in),...,fm(i1,...,in)) = ...
S2                ... = A(g1(i1,...,in),...,gm(i1,...,in))
        ENDDO
        ...
  ENDDO
ENDDO
```

Iterativni vektor predstavlja uređenu n-torku indeksa: $I=(i_1,i_2,..,i_n)$
Ako su petlje zadate u normalizovanom obliku, onda su iterativni vektori prirodno uređeni u *leksikografskom* redosledu.
Zavisnost između instrukcija S1 i S2 postoji, ako postoje dva iterativna vektora, α i β, takva da je

- $\alpha < \beta$, i
- $f_i(\alpha) = g_i(\beta)$ **,** za svako i, $1 \leq i \leq m$

- Usvajamo da us f_i i g_i, affine funkcije
- Da bi utvrdili da li postoji zavisnost potrebno je rešiti sistem od m Diofantovih jednačina sa n nepoznatih.
- To je np kompleksan problem! Zbog toga se primenjuju tvz. konzervativni testovi.
  - usvaja se da zavisnost postoji, osim ako se dokaže da ne postoji!

Definicije pojmova

- Indeks – iterativna promenljiva petlje
- subscript – pozicija (indeks,dimenzija) u referenci polja (array)

```
for(i = 0; i < 10; i++)
for(j = 0; j < 10; j+=2)
for(k = 10; k > 0; k--)
A[0][i][j] = A[10][i][k];
```

- i,j, k su indeksi
- [0], [i], [j], [10], ... su subscript
- subscript može biti konstanta, indeks ili linearna kombinacija indeksa
  - subscripti se uvek posmatraju u paru
  - parovi: <0,10>; <i,i>; <j,k>

Koliko ima indeksa u sabscript paru?

- Broj indeksa u sbscript paru se zove kompleksnost:
  - postoje tri nivoa kompleksnosti koji se koriste kod analize zavisnosti
  - ZIV –zero index variable
    - subscript par ne sardži iterativne promenljive u paru
    - `A[0][i][j] = A[10][i][k];`

    Prvi subscript par je ZIV. U prvoj indeksnoj poziciji nalaze se samo konstante: <0,10>
  - SIV –single index variable
    - subscript par sadrži samo jednu iterativnu promenljivu
    - `A[0][i][j] = A[10][i][k];`

    drugi subscript par je SIV: <i,i>
  - MIV –multiple index variable
    - subscript par sadrži više iterativnih promenljivih
    - treći subscript par je MIV: <j,k>

Separabilni i povezani subscripti

- Subscript je separabilan ako se iterativne promenljive koje se nalaze u njemu ne javljaju u drugim subscriptima.
  - a(i+1,j)=a(k,j)+c
    - oba subscripta su separabilna
- ako dva različita subscripta sadrže iste indekse, onda su oni povezani (coupled)
  - a(i,j,j)=a(i,j,k)
    - drugi i treći subscripti su povezani (iterativna promenljiva j se javlja i u drugoj i u trećoj poziciji u referenci polja)
    - povezani subscripti komplikuju analizu zavisnosti

Analiza zavisnosti

- Prvi korak u analizi zavisnosti je da se klasifikuju subscripti kao ZIV / SIV / MIV i separabilne i povezane
- Zašto je to bitno?
  - pojednostavljuje testiranje na zavisnost
    - za separabilne subscripte se mogu dobiti egzaktni odgovori na pitanje da li postoji zavisnost
  - Ako pokažemo za bilo koji subscript da ne postoji rešenje, tj ne postoje dva iterativna vektora α i β, α< β, tako daje $f_i(\alpha)=g_i(\beta)$, dokazali smo da nema zavisnosti između dva referenciranja nekom elementu nekog polja.
- Primer:

```
for(i = 0; i < 10; i++)
for(j = 0; j < 10; j+=2)
for(k = 10; k > 0; k--)
A[0][i][j] = A[10][i][k];
```

- Prvi subsript par je <0,10> i pošto je $0 \neq 10$, ne postoji zavisnost u petlji.

[redacted]

- Prvo tražimo da li postoji ZIV subscript
  - A[..., cw , ...] = A[..., cr, ...]
    - ako su cw i cr konstante i cw≠cr, sigurno nema zavisnosti!
  - Primer
    - A[5, j+1, 10, k] = A[i, j, 12, k-1] + c
    - treći subscript je ZIV, i pošto je 10 ≠12, sigurno nema zavisnosti
- Ako nema ZIV ili nam ZIV ne da odgovor, pronalaze se SIV subscripti
  - primenjujemo GCD test
    - ako test kaže da nema zavisnosti u datom subscript paru, zavisnost ne postoji!

Vektori zavisnosti i vektori pravca zavisnosti

- Umesto rešavanja sistema Diofantovih jednačina, nekada je moguće otkriti zavisnosti u gnezdu petlji pomoću vektora zavisnosti
  - Da bi uočili kakve zavisnosti postoje izmedju naredbi unutar tela petlje potrebno je prvo uočiti sve parove generisanih—korišćenih promenljivih i za svaki takav par odrediti vektor zavisnosti d.
    - Od svih vektora zavisnosti formira se matrica zavisnosti po podacima, D
    - Ako je generisana promenljiva X(f(I)), gde je f celobrojna funkcija definisana nad indeksnim skupom I, a X(g(I)) korišćena promenljiva (g je opet celobrojna funkcija definisana nad indeksnim skupom I), vektor zavisnosti se izračunava kao

$$d=f(I) - g(I).$$

Ako su d_1, d_2, ... , d_k svi vektori zavisnosti, tada se matrica zavisnosti po podacima dobija kao

$$D = [\ d_1\ \ d_2\ \dots\ d_k]$$

PRIMER

- Pronaći sve vektore zavisnosti u sledećem gnezdu petlji

```
    for  i = 1, 5
    for  j = 1, 10
    fot  k =1, 20
  A(i, j, k) = A(i-1, j, k+1)+ B(i, j, k)
    B(i, j, k+1) = B(i, j-1, k-1) * 3
  endfor{i,j,k}
```

$$D = \begin{bmatrix} d_1 & d_2 & d_3 \end{bmatrix} = \begin{array}{c} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & 2 \end{bmatrix} \end{array} \begin{array}{c} i \\ j \\ k \end{array}$$

matrica zavisnosti

- REŠENJE:
  - Uočimo prvo sve parove generisanih—korišćenih promenljivih:

  A(i, j, k) i A(i-1, j, k+1)

  B(i, j, k+1) i B(i, j, k),

  B(i, j, k+1) i B(i, j-1, k-1)

Odgovarajući vektori zavisnosti

$d_1 = (i, j, k)^T - (i-1, j, k+1)^T = [\,1, 0, -1]^T$

$d_2 = (i, j, k+1)^T - (i, j, k)^T = [\,0, 0, 1]^T$

$d_3 = (i, j, k+1)^T - (i, j-1, k-1)^T = [0, 1, 2]^T$

- Prvi element u vektoru koji je ≠0 nosi zavisnost!

vektorizacija ugnježdjenih petlji (nast.)

- Često je za analizu zavisnosti po podacima dovoljno poznavati samo pravce zavisnosti, a ne stvarne vrednosti vektora zavisnosti.
- Pravac zavisnosti se definiše na sledeći način:

$$pravac\_zavisnosti = \begin{cases} <, & if \quad d_i > 0 \\ =, & if \quad d_i = 0 \\ >, & if \quad d_i < 0 \end{cases}$$

- Za prethodne vektore zavisnosti odgovarajući pravci zavisnosti su
  - $d_1 = [\ 1, 0, -1]^T \quad \Rightarrow \quad [\ <, =, >\ ]^T$
  - $d_2 = [\ 0, 0, 1]^T \quad \Rightarrow \quad [\ =, =, <\ ]^T$
  - $d_3 = [0, 1, 2]^T \quad \Rightarrow \quad [\ =, <, <\ ]^T$

$$D = \begin{array}{c} \begin{bmatrix} < & = & = \\ = & = & < \\ > & < & < \end{bmatrix} \end{array} \begin{array}{c} i \\ j \\ k \end{array}$$

Pravilo

- Zavisnost nosi prvi element vektora koji je $<$ ili $>$.
- Pravac $=$ ne sprečava vektorizaciju.

$$D = \begin{array}{cc} \begin{bmatrix} < & = & = \\ = & = & < \\ > & < & < \end{bmatrix} & \begin{array}{c} i \\ j \\ k \end{array} \end{array}$$

- vektorizacija po indeksnoj promenljivoj *k* nije moguća jer za drugi vektor zavisnosti postoji looop-carry po indeksnoj promenljivoj k

Vektorizacija ugnj. petlji (nast.)

- Da zaista postoji loop-carry zavisnost koja sprečava vektorizaciju po indeksnoj promenljivoj k, možemo se uveriti ako izvršimo odmotavanje petlje po *k* za neku fiksnu vrednost promenljivih *i* i *j*.

✱ na primer, za i=1 i j=1 i k=1, 2,....,20 imamo

- i=1, j=1, k=1 A(1, 1, 1) = A(0, 1, 2) + B(1, 1, 1)
- B(1, 1, 2) = B(1, 0, 0) *3
- loop-carry zavisnost
- i=1, j=1, k=2 A(1, 1, 2) = A(0, 1, 3) + B(1, 1, 2)
- B(1, 1, 3) = B(1, 0, 1) *3

loop-carry zavisnost

- i=1, j=1, k=3 A(1, 1, 3) = A(0, 1, 4) + B(1, 1, 3)
- B(1, 1, 4) = B(1, 0, 2) * 3

Primer2

- Nekada rastojanje između iteracija nije konstantno

```
for (j =1; j<=10; j++){
    for (i=1; i<=99; i++){
S1: A(i,j)=B(i,j)+x;
S2: C(i,j)= A(100-i,j) +y
}
}
```

- Vektor zavisnosti: $d=(i,j)^T-((100-i,j)^T= (2i-100,0)^T$
  - da li je 2i-100 =0, >0 ili <0?
  - distanca nije konstanta već se menja
    - za i < 50, 2i-100 <0
    - za i=50, 2i-100=0
    - za i >50, 2i-100 >0
  - Moramo uzeti sve u obzir pa je vektor pravca zavisnosti d =(=,*)
    - u suštini imamo tri vektora (=, <), (=,=) i (=, >)

Primer 3

- Šta ako se neka indeksna promenljiva ne pojavljuje ni i jednom subscriptu?

```
for (i=0; i<100; i++)
 for (j=0; j<100; j++)
        a(i+1)= a(i)+B(j);
```

  - Moramo uzeti u obzir sve mogućnosti za nedostajući indeks
    - d =(1,*), tj d=(<,*)
    - $D = \begin{array}{cc} \begin{bmatrix} < & < & < \\ < & = & > \end{bmatrix} & \begin{array}{c} i \\ j \end{array} \end{array}$

Vektorizacija ugnj. petlji (nast.)

- Ako vektorizacija nije moguća po unutrašnjoj petlji, može se pokušati sa zamenom petlji ili nekim drugim transformacijama indeksnih promenljivih.
- Postoje tri elementarne transformacije koje se mogu obavljati nad indeksnim skupovima, tj. nad petljama.
  - Ove transformacije se opisuju pomoću matrica transformacija, T.
- Ove matrice moraju da poseduju sledeće tri osobine:
  1. To su kvadratne matrice, što znači da vrše preslikavanje n-dimenzionalnog indeksnog prostora u n-dimenzionalni indeksni prostor
  2. To su celobrojne matrice
  3. $|\det T| = 1$

Vektorizacija ugnj. petlji (nast.)

- Zbog ovih osobina proizvod dve elementarne transformacije daje važeću transformaciju.
- Da bi jedna transformacija mogla da se primeni nad indeksnim skupom a da to ne utiče na korektnost izračunavanja, matrica transformacije T ne sme da menja znak vektora zavisnosti:
  - Ako je *d* vektor zavisnosti, T matrica transformacije, tada novi vektor zavisnosti $\hat{d}$, koji se dobija kada se T primeni na *d*, tj.

$$\hat{d} = T \cdot d$$

  - mora biti istog znaka kao i *d*.
    - Ako je d>0, tada mora i $\hat{d}$ >0, ili ako je d < 0, tada i $\hat{d}$ mora <0.
    - Za vektor se kaže da je pozitivan (negativan) ako mu je prvi ne-nulti element pozitivan (negativan).

Vektorizacija ugnj. petlji (nast.)

- Vektor
- $$d = \begin{bmatrix} 0 \\ 1 \\ -2 \end{bmatrix}$$ je > 0
- $$d = \begin{bmatrix} 0 \\ -1 \\ 2 \end{bmatrix}$$ je < 0

Elementarne transformacije nad indeksnim skupovima

1. *permutacija*
2. *obrtanje redosleda*
3. *krivljenje*

Permutacija – omogućava zamenu mesta dvema petljama

$$
\begin{aligned}
I_1 &= l_1, u_1 \\
I_2 &= l_2, u_2 \\
&\vdots \\
I_m &= l_m, u_m
\end{aligned}
$$

Ako želimo da zamenimo mesta petljama po indeksima I_j i I_k, ta transformacija se opisuje pomoću matrice T koja se dobija kada se u jediničnoj matrici zamene mesta j-toj i k-toj vrsti.

Transformacija permutacije - primer

- za m=2 matrica transformacije T je oblika

$$T = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Ako smo imali petlje

    I = 1, n

      J = 1, m

nakon transformacije T dobićemo

    J= 1, m

      I = 1, n

Transformacija permutacije

- PRIMER: Izvršiti permutaciju indeksnih promenljivih I i J u sledećem gnezdu petlji

```
for I = 1, n
  for J = 1, n
 A(J) = A(J) + C(I, J)
endfor{I,J}
```

- Primenom transformacije permutacije nad indeksnim skupom $(I, J)^T$ dobijamo nove indeksne promenljive U i V na sledeći način

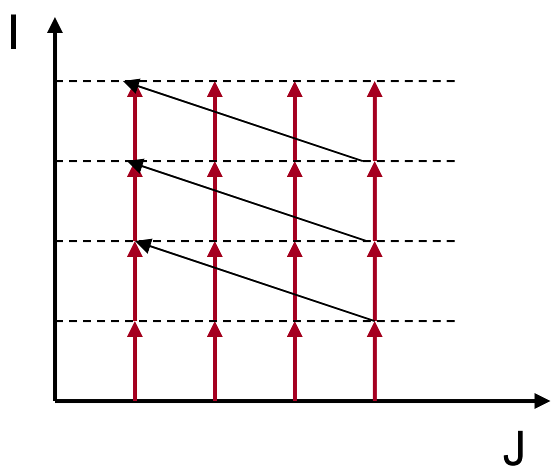
$$\begin{bmatrix} U \\ V \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} I \\ J \end{bmatrix} = \begin{bmatrix} J \\ I \end{bmatrix}$$

- pri čemu je U = J, a V = I.
- Granice za U i V odredjujemo na osnovu granica za indeksne promenljive J i I, respektivno.
- Transformisano gnezo petlji sada ima oblik

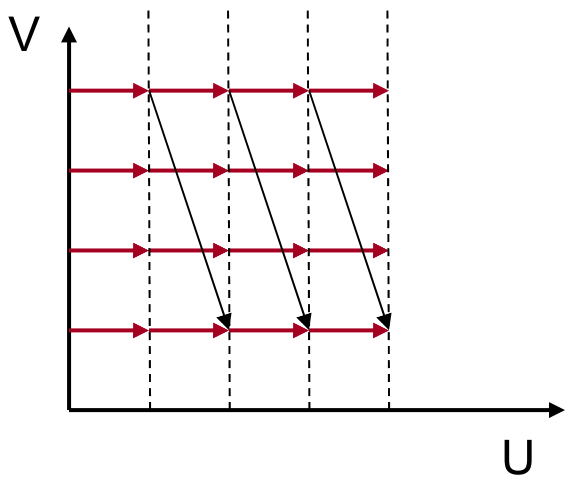
```
for 10  U = 1, n
  for 10  V = 1, n
 A(U) = A(U) + C(V, U)
endfor{u,v}
```

Transformacija permutacije

- Transformacijom je promenjen redosled izračunavanja elemenata vektora A.



Pre transformacije



Posle transformacije

Da li je legalno izvšiti permutaciju?

```
for (i = 0; i < 100; ++i)
    for (j = 0; j < 100; ++j)
        a[i+1] = a[i] + B[j];
```

$$\vec{D}_a = [<, *]$$

```
for (j = 0; j < 100; ++j)
    for (i = 0; i < 100; ++i)
        a[i+1] = a[i] + B[j];
```

$$\vec{D}_a = [*, <]$$

$$\vec{D}_a = \{[<, <]\ [=, <]\ [>, <]\}$$

- $[<, <]$ is a level-1 true dependence
- $[=, <]$ is a level-2 true dependence
- $[>, <]$ is a level-1 anti-dependence. It exists because of the following iteration numbers: $j = 0$ and $i = 1$, and $j = 1$ and $i = 0$

Transformacija obrtanje

- Omogućava promenu redosleda izračunavanja po odredjenoj indeksnoj promenljivoj
- Transformacija se opisuje jediničnom matricom u kojoj je u i-toj vrsti dijagonalni element jednak –1.

- PRIMER: Posmatrajmo ovakvo gnezdo petlji:

```
for i = 1, n
for j = 1, n
  A(i, j) = A(i-1, j-2)*k
 endfor{i,j}
```

- Želimo da primenimo transformaciju obrtanja po indeksnoj promenljivoj *j*.
- Matrica transformacije je oblika

$$T = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Obrtanje – primer (nast.)

- Novi indeksni skup dobijamo na sledeći način

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} i \\ j \end{bmatrix} = \begin{bmatrix} i \\ -j \end{bmatrix}$$

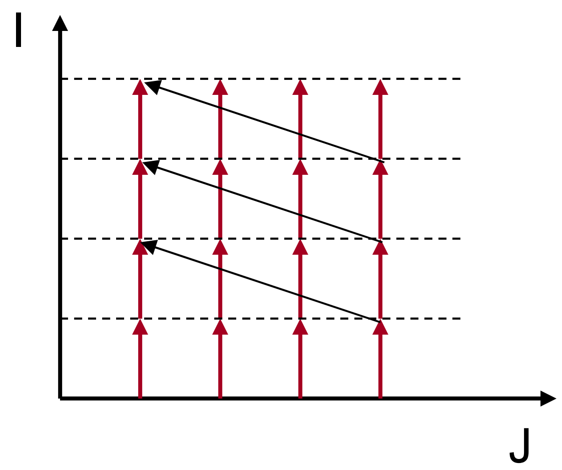
- Što znači da je $u = i$, $v = -j$, a granice indeksa u i v su u=1, n i v=-n, -1.

- Transformisana petlja ima sledeći izgled

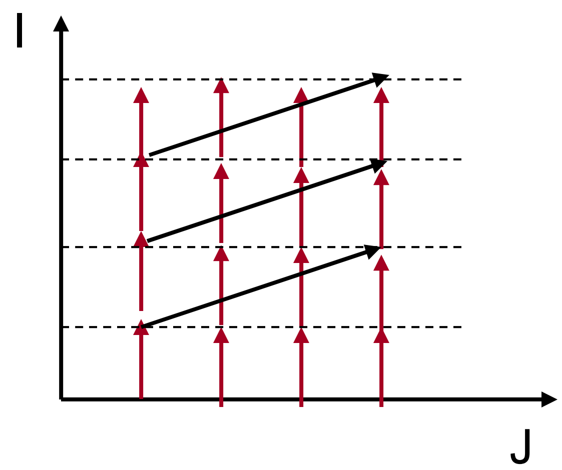
```
for u = 1, n
for v = - n, -1
A(u, -v) = A(u-1, -v-2)*k
endfor{u,v}
```

- Transformacijom je izmenjen redosled izračunavanja po indeksnoj promenljivoj *j*

Obrtanje – primer



Pre transformacije



Posle transformacije

- Npr. za n=3 i i=1 pre transformacije se redom izračunavaju elementi
  A(1,1), A(1,2) i A(1,3),
- Nakon transformacije redosled izračunavanja elemenata je

  A(1,3), A(1,2) i A(1,1).

3. Transformacija krivljenja (skewing)

- Ovom transformacijom se obavlja krivljenje (skeweing) jednog iterativnog indeksa u odnosu na drugi za faktor **f**.
- Pretpostavimo da imamo ovakvu iteraciju indeksnih promenljivih

$$(p_1, p_2, \ldots, p_i, \ldots, p_{j-1}, p_j, p_{j+1}, \ldots, p_n)$$

- ako primenimo krivljenje petlje I_j u odnosu na I_i za faktor f izvršiće se preslikavanje gornje iteracije u

$$(p_1, p_2, \ldots, p_i, \ldots, p_{j-1}, p_j + f \cdot p_i, p_{j+1}, \ldots, p_n)$$

Krivljenje

- Krivljenje petlje i_k u odnosu na i_j za faktor f

$$
\begin{bmatrix}
1 & 0 & .. & & & & .. & 0 & 0 \\
0 & 1 & .. & & & & .. & 0 & 0 \\
\vdots & & & & & & & \vdots & \vdots \\
0 & 0 & .. & 1 & .. & & .. & 0 & 0 \\
\vdots & & & & & & & \vdots & \vdots \\
0 & 0 & .. & f & .. & 1 & .. & 0 & 0 \\
\vdots & & & \vdots & & & .. & \vdots & \vdots \\
0 & 0 & .. & & & & .. & 1 & 0 \\
0 & 0 & .. & & & & .. & 0 & 1
\end{bmatrix}
\begin{bmatrix}
i_1 \\ i_2 \\ \\ ij \\ \\ i_k \\ \\ \\ i_n
\end{bmatrix}
=
\begin{bmatrix}
i_1 \\ i_2 \\ \\ ij \\ \\ i_k + fij \\ \\ \\ i_n
\end{bmatrix}
$$

Krivljenje – primer

- PRIMER: Posmatrajmo sledeće gnezdo petlji

```
for i = 1, n
for j = 1, n
  A(i, j) = A(i, j-1) + A(i-1,j)
endfor{i,j}
```

- Primenimo krivljenje indeksne promenljive *j* u odnosu na *i* za faktor 2.
- Transformacija se opisuje na sledeći način $T = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}$
- Novi indeksni skup je $\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} i \\ j \end{bmatrix} = \begin{bmatrix} i \\ j+2i \end{bmatrix}$
- što znači da je $u=i$, $v=2i+j$,
- granice novih indeksnih promenljivih su $u=1,n$ i $v=2u+1,2u+n$.

* Matrica krivljenja

$$T = \begin{bmatrix} 1 & 0 \\ q & 1 \end{bmatrix}$$

Original Loop

```
for I1 = n1, N1
  for I2 = n2, N2
     H(I1, I2)
  end for
end for
```

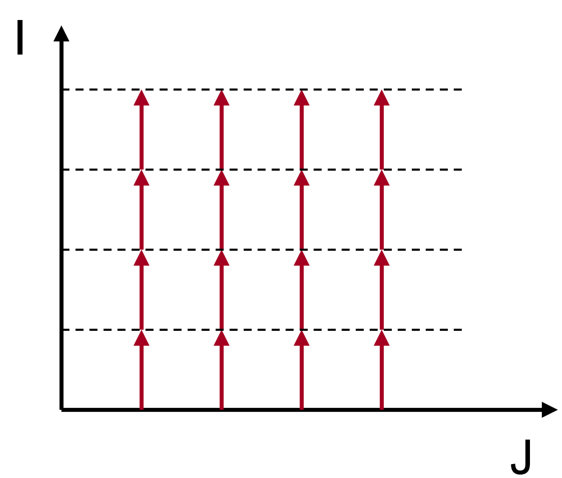
Transformed Loop

```
for K 1 = n1, N1
  for K 2 = n2 + q*K 1, N2 + q*K 1
          H(K 1, K 2 - q*K 1)
  end for
end for
```

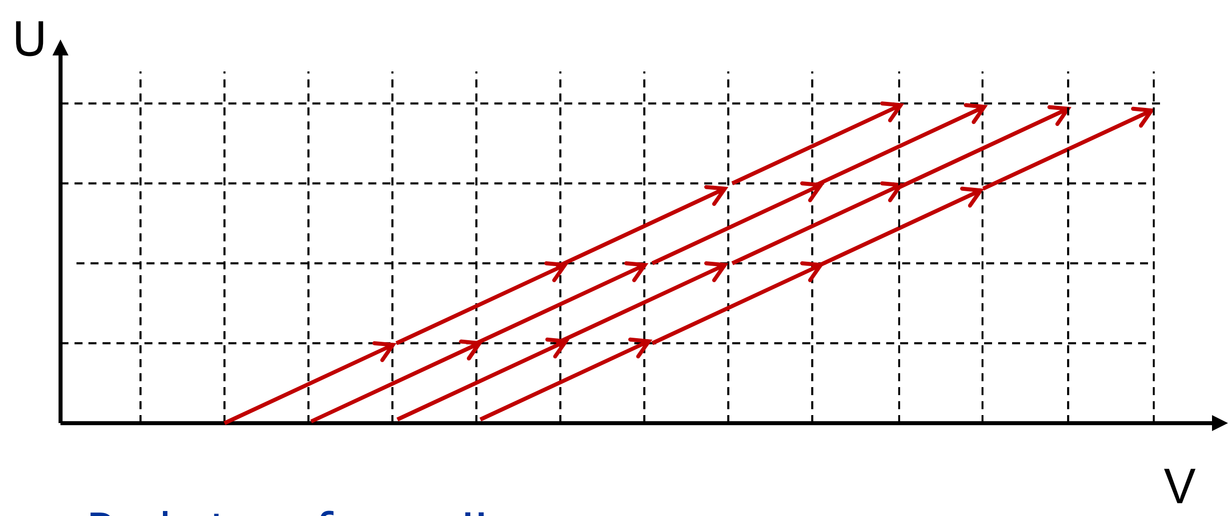
Krivljenje – primer (nast.)

- Transformisana petlja ima sledeći izgled

```
for u = 1, n
for v = 2u+1, 2u+ n
A(u, v-2u) = A(u,v-2u-1) + A(u-1, v-2u)
endfor{u,v}
```



Pre transformacije



Posle transformacije

Primer-1

- Da bi neka transformacija mogla da se primeni nad indeksnim skupom ona ne sme da menja znak vektora zavisnosti da bi se sačuvale zavisnosti koje postoje u redosledu izračunavanja

```
    for  i = 1, 5
    for  j = 1, 10
    for  k =1, 20
A(i, j, k) = A(i-1, j, k+1)+ B(i, j, k)
    B(i, j, k+1) = B(i, j-1, k-1) * 3
endfor{i,j,k}
```

$$D = [d_1 \quad d_2 \quad d_3] = \begin{array}{c} \\ \\ \\ \end{array}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & 2 \end{bmatrix}\begin{array}{c} i \\ j \\ k \end{array}$$

matrica zavisnosti

- REŠENJE:
  - Uočimo prvo sve parove generisanih—korišćenih promenljivih:

  A(i, j, k) i A(i-1, j, k+1)

  B(i, j, k+1) i B(i, j, k),

  B(i, j, k+1) i B(i, j-1, k-1)

Odgovarajući vektori zavisnosti

$d_1 = (i, j, k)^T - (i-1, j, k+1)^T = [\,1, 0, -1]^T$

$d_2 = (i, j, k+1)^T - (i, j, k)^T = [\,0, 0, 1]^T$

$d_3 = (i, j, k+1)^T - (i, j-1, k-1)^T = [0, 1, 2]^T$

- Zbog vektora d2 nije moguće izvršiti vektorizaciju.
  - permutacijom petlji j i k dobićemo kod koji je moguće vektorizovati

Primer-1

- Matrica zavisnosti pre permutacije

$$D = \begin{bmatrix} d_1 & d_2 & d_3 \end{bmatrix} = \begin{array}{c} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & 2 \end{bmatrix} \end{array} \begin{array}{c} i \\ j \\ k \end{array}$$

- Matrica zavisnosti nakon permutacije

$$D = \begin{bmatrix} d_1 & d_2 & d_3 \end{bmatrix} = \begin{array}{c} \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \begin{array}{c} i \\ k \\ j \end{array}$$

- zavisnosti su zadržane ali se unutrašnja petlja može vektorizovati

Primer-1–nast.

- transformisana petlja

```
for  i = 1, 5
for  k = 1, 20
for  j = 1, 10
A(i, j, k) = A(i-1, j, k+1)+ B(i, j, k)
      B(i, j, k+1) = B(i, j-1, k-1) * 3
endfor{i,j,k}
```

- i=1,k=1,j=1
  - a(1,1,1)=a(0,1,2)+b(1,1,1)
  - b(1,1,2) =b(1,0,0)*3
- i=1,k=1,j=2
  - a(1,2,1)=a(0,2,2)+b(1,2,1)
  - b(1,2,2) =b(1,1,0)*3

⋮

- i=1,k=2,j=1
  - a(1,1,2)=a(0,1,3)+b(1,1,2)
  - b(1,2,3) =b(1,0,1)*3

Primer2

- Da bi neka transformacija mogla da se primeni nad indeksnim skupom ona ne sme da menja znak vektora zavisnosti da bi se sačuvale zavisnosti koje postoje u redosledu izračunavanja.
  - PRIMER: Ako imamo ovakvo jedno gnezdo petlji
  - for i = 1, 3
  - for j = 1, 2
  - A(i, j) = A(i-1, j+1) *2
  - endfor{i,j}
  - vektor zavisnosti je $d=(i, j)^T - (i-1, j+1)^T = (1, -1)^T > 0$.
  - Ako bismo primenili transformaciju permutacije, narušili bismo zavisnosti koje postoje u redosledu izračunavanja jer je

$$\hat{d} = T \cdot d = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} < 0$$

  - a vektor d je >0!

Primer2 – nast.

```
for i = 1, 3
for j = 1, 2
A(i, j) = A(i-1, j+1) *2
endfor{i,j}
```

za i=1, j=1 izračunava se A(1,1)=A(0,2)*2
j=2 izračunava se A(1,2)=A(0,3)*2

za i=2, j=1 izračunava se A(2,1)=A(1,2)*2
j=2 izračunava se A(2,2)=A(1,3)*2

za i=3, j=1 izračunava se A(3,1)=A(2,2)*2
j=2 izračunava se A(3,2)=A(2,3)*2

- Strelicama je označen redosled zračunavanja koji mora biti ispoštovan:
  - A(1, 2) mora biti izračunat pre A(2,1) jer A(2,1) koristi vrednost A(1,2).
  - A(2,2) mora biti izračunat pre A(3,1)

- Ako na prethodnu petlju primenimo transformaciju permutacije dobićemo $\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} i \\ j \end{bmatrix} = \begin{bmatrix} j \\ i \end{bmatrix}$

```
for u = 1, 2
for v = 1, 3
A(v, u) = A(v-1, u+1) *2
enfor{u,v}
```

za u=1, v=1 izračunava se A(1,1)=A(0,2)*2
v=2 izračunava se A(2,1)=A(1, 2)*2
v=3 izračunava se A(3,1)=A(2, 2)*2

za u=2, v=1 izračunava se A(1,2)=A(0,3)*2
v=2 izračunava se A(2,2)=A(1, 3)*2
v=3 izračunava se A(3,2)=A(2, 3)*2

- Prvo izračunava element A(2,1) pa nakon toga element A(1,2), što je pogrešno!

Kompozicija transformacija

- Zbog osobina elementarnih matrica transformacija, proizvod elementarnih matrica transformacija daje takodje validnu transformaciju.
  - Tako, da bi u prethodnom primeru mogli da primenimo permutaciju, a da ne narušimo zavisnosti po podacima, možemo da primenimo kompoziciju transformacija permutacije i obrtanja:

$$T = T_{\text{permutacija}} T_{\text{obrtanje}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

  - Novi vektor zavisnosti biće pozitivan, tj.

$$\hat{d} = T \cdot d = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} > 0$$

Primer (nast.)

for i = 1, 3
for j = 1, 2
A(i, j) = A(i-1, j+1) *2
endfor{i,j}

za i=1, j=1 izračunava se A(1,1)=A(0,2)*2
j=2 izračunava se A(1,2)=A(0,3)*2

za i=2, j=1 izračunava se A(2,1)=A(1,2)*2
j=2 izračunava se A(2,2)=A(1,3)*2

za i=3, j=1 izračunava se A(3,1)=A(2,2)*2
j=2 izračunava se A(3,2)=A(2,3)*2

Kompozicija transformacija permutacija+obrtanje

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} i \\ j \end{bmatrix} = \begin{bmatrix} -j \\ i \end{bmatrix}$$

- transformisana petlja

for u=-2,-1
for v=1,3
A(v,-u)=A(v-1,-u+1)*2
endfor{u,v}

za u=-2, v=1 izračunava se A(1,2)=A(0,3)*2
v=2 izračunava se A(2,2)=A(1,3)*2
v=3 izračunava se A(3,2)=A(2,3)*2

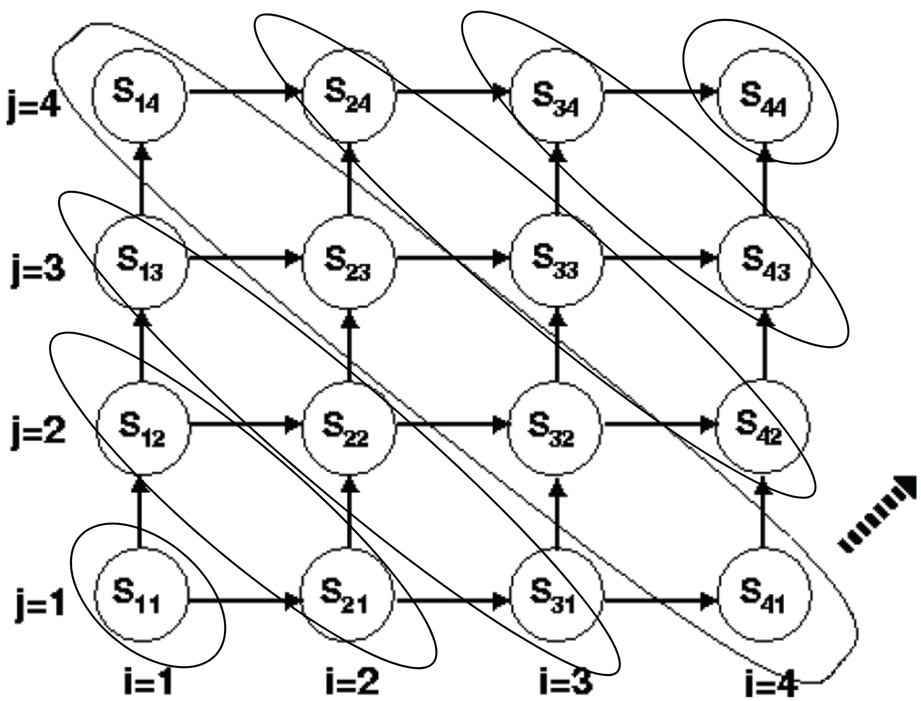
za u=-1, v=1 izračunava se A(1,1)=A(0,2)*2
v=2 izračunava se A(2,1)=A(1,2)*2
v=3 izračunava se A(3,1)=A(2,2)*2

Redosled izračunavanja je ispoštovan!

Primer

- Da li se sledeća petlja može vektorizovati?

```
for i = 1:N
for j = 1:M
A(i,j) = A(i-1,j) + A(i,j-1);
```



Krivljenje?
permutacija?