Dependence Analysis

For Double Loop

Delivered by

Dr. Jitendra V. Tembhurne

Index and Iteration Space

- Index space:- Set of all Index point
- Iteration space:- Set of all iteration point

```
do I1= p1,q1,Θ1
    do I2= p2,q2, Θ2
        H(I1,I2)
    enddo
enddo
```

We assume that the loop parameters are integer-valued; p_1, q_1, θ_1, and θ_2 are constants; θ_1 and θ_2 are nonzero; and p_2 and q_2 are affine functions of I_1. Let

$$p_2 = p_{20} + p_{21} I_1$$
$$q_2 = q_{20} + q_{21} I_1,$$

Theorem 3.1: *The iteration space of the double loop* (L1,L2) *consists of all integer vectors* $(\hat{I}_1, \hat{I}_2)$ *such that*

$$0 \le \hat{I}_1 \le \hat{q}_1$$
$$0 \le \hat{I}_2 \le \hat{q}_{20} + \hat{q}_{21} \hat{I}_1,$$

$$\hat{q}_1 = \lfloor (q_1 - p_1)/\theta_1 \rfloor$$
$$\hat{q}_{20} = [(q_{20} - p_{20}) + (q_{21} - p_{21})p_1]/\theta_2$$
$$\hat{q}_{21} = (q_{21} - p_{21})\theta_1/\theta_2.$$

- *The index space of* (L1,L2) *is the set of all integer vectors* (I_1,I_2) *given by*

$$I_1 = p_1 + \theta_1 \hat{I}_1$$
$$I_2 = (p_{20} + p_{21}p_1) + (p_{21}\theta_1)\hat{I}_1 + \theta_2 \hat{I}_2,$$

$\hat{L}_1$: **do** $\hat{I}_1 = 0, \hat{q}_1, 1$

$\hat{L}_2$: **do** $\hat{I}_2 = 0, \hat{q}_{20} + \hat{q}_{21}\hat{I}_1, 1$

$\hat{H}(\hat{I}_1, \hat{I}_2)$

enddo

enddo

Issues related to double loop

- A double loop (L1,L2) with arbitrary stride can be written as an equivalent double loop with unit stride.
- p2 and q2 are affine function of I1

Advantage:- Index space is traversed in the direction of lexicographically increasing.

Example

$L_1:$ **do** $I_1 = 20, 11, -2$

$L_2:$ **do** $I_2 = I_1 + 2, 2I_1 + 1, 3$

$H(I_1, I_2)$

enddo

enddo

I_1 takes the values: $20, 18, 16, 14, 12,$

$$I_1 = 12, I_1 = 20, I_2 = I_1 + 2, I_2 = 2I_1 + 1.$$

I_1				I_2	$\rightarrow$		
$\downarrow$ 20	22	25	28	31	34	37	40
18	20	23	26	29	32	35	
16	18	21	24	27	30	33	
14	16	19	22	25	28		
12	14	17	20	23			

$\hat{I}_1$				$\hat{I}_2$	$\rightarrow$		
$\downarrow$ 0	0	1	2	3	4	5	6
1	0	1	2	3	4	5	
2	0	1	2	3	4	5	
3	0	1	2	3	4		
4	0	1	2	3			

It is traversed in the direction:

$$(20,22),\ldots,(20,40),(18,20),\ldots,(18,35),\ldots,(12,14),\ldots,(12,23).$$

The corresponding sequence of iteration points is

$$(0,0),\ldots,(0,6),(1,0),\ldots,(1,5),\ldots,(4,0),\ldots,(4,3).$$

Next, apply Theorem 3.1 and its corollary to this example. We have

$$\begin{aligned} p_1 &= 20, & p_{20} &= 2, & q_{20} &= 1, \\ q_1 &= 11, & p_{21} &= 1, & q_{21} &= 2, \\ \theta_1 &= -2, & & & \theta_2 &= 3. \end{aligned}$$

$$\hat{q}_1 = 4, \quad \hat{q}_{20} = 19/3, \quad \hat{q}_{21} = -2/3.$$

$$\begin{aligned} &0 \le \hat{I}_1 \le 4 \\ &0 \le \hat{I}_2 \le 19/3 - 2\hat{I}_1/3. \end{aligned}$$

Thus, the iteration space is bounded by the lines:

$$\hat{I}_1 = 0, \hat{I}_1 = 4, \hat{I}_2 = 0, 2\hat{I}_1 + 3\hat{I}_2 = 19.$$

The index and iteration spaces are related by equations

$$I_1 = 20 - 2\hat{I}_1$$
$$I_2 = 22 - 2\hat{I}_1 + 3\hat{I}_2.$$

When the loops are normalized, we get the equivalent program

$\hat{L}_1$: **do** $\hat{I}_1 = 0, 4, 1$
$\hat{L}_2$: **do** $\hat{I}_2 = 0, (19 - 2\hat{I}_1)/3, 1$
$H(20 - 2\hat{I}_1, 22 - 2\hat{I}_1 + 3\hat{I}_2)$
enddo
enddo

Dependence in double loop

Consider two assignment statements S and T in the loop nest (L1, L2). Let (d1, d2) denote the distance from an instance S (i1, i2) of S to an instance T(j1,j2) of T.

In the sequential execution of the program, S(i1,i2) is executed before T(j1, j2) iff one of the following two conditions holds:

- (d1,d2) < (0,0),
- (d1, d2) = (0, 0) and S < T.

Necessary condition for dependence

- For two statements S and T, we have S ∂ T if there is an instance *S(i1, i2) of S, an instance T(jl,j2) of T, and a* memory location M, such that

1. Both S(i1, i2) and *T(jl,j2) reference (read or write) M;*
2. S(i1, i2) is executed before *T(j1, J2) in the sequential execution* of the program;
3. During sequential execution, the location M is not written in the time period from the end of execution of S(i1, i2) to the beginning of the execution of T(j1, j2).

- From the above conversion and calculation we can find our all dependencies exists in the code segment and we can find distance and direction vector also.

Dependence Demo Calculation

Example:-

L1 : do $I_1 = 0, 100, 1$

L2 : do $I_2 = 0, 50, 1$

S: $X(3I_1, 3I_2) = \ldots.$

T: $\ldots. \qquad = X(2I_1 + 1, I_2 + 2)$

enddo

enddo

Dependence Distance Vector

Definition:

Suppose that there is a dependence from S on iteration $\boldsymbol{i}$ to T on iteration $\boldsymbol{j}$; then the *dependence distance vector* $\mathbf{d}(\boldsymbol{i}, \boldsymbol{j})$ is defined as

$$\boldsymbol{d}(\boldsymbol{i}, \boldsymbol{j}) = \boldsymbol{j} - \boldsymbol{i}$$

Example:

```
  DO I = 1, 3
    DO J = 1, I
S    A(I+1,J) = A(I,J)
    ENDDO
  ENDDO
```

True dependence between S and itself on

$\boldsymbol{i} = (1,1)$ and $\boldsymbol{j} = (2,1)$: $\boldsymbol{d}(\boldsymbol{i}, \boldsymbol{j}) = (1,0)$

$\boldsymbol{i} = (2,1)$ and $\boldsymbol{j} = (3,1)$: $\boldsymbol{d}(\boldsymbol{i}, \boldsymbol{j}) = (1,0)$

$\boldsymbol{i} = (2,2)$ and $\boldsymbol{j} = (3,2)$: $\boldsymbol{d}(\boldsymbol{i}, \boldsymbol{j}) = (1,0)$

Dependence Direction Vector

Definition:

Suppose that there is a dependence from S on iteration $\boldsymbol{i}$ and T on iteration $\boldsymbol{j}$; then the *dependence direction vector* $\mathbf{D}(\boldsymbol{i}, \boldsymbol{j})$ is defined as

$$\boldsymbol{D}(\boldsymbol{i}, \boldsymbol{j}) \text{ or } \mathbf{D(d1,d2)} = \begin{cases} \text{“<”} & \text{if } \boldsymbol{d}(\boldsymbol{i}, \boldsymbol{j}) > 0 \\ \text{“=”} & \text{if } \boldsymbol{d}(\boldsymbol{i}, \boldsymbol{j}) = 0 \\ \text{“>”} & \text{if } \boldsymbol{d}(\boldsymbol{i}, \boldsymbol{j}) < 0 \end{cases}$$

Where , d1=i1-i2 & d2 = j1-j2

By using direction vector, we can find the level of dependence i.e

1 → inner loop

2 → outer loop

3 → involve both loops

Example

```
  DO I = 1, 3
    DO J = 1, I
S    A(I+1,J) = A(I,J)
    ENDDO
  ENDDO
```

True dependence between S and itself on

$\boldsymbol{i} = (1,1)$ and $\boldsymbol{j} = (2,1)$: $\boldsymbol{d}(\boldsymbol{i}, \boldsymbol{j}) = (1,0)$, $\boldsymbol{D}(\boldsymbol{i}, \boldsymbol{j}) = (<, =)$
$\boldsymbol{i} = (2,1)$ and $\boldsymbol{j} = (3,1)$: $\boldsymbol{d}(\boldsymbol{i}, \boldsymbol{j}) = (1,0)$, $\boldsymbol{D}(\boldsymbol{i}, \boldsymbol{j}) = (<, =)$
$\boldsymbol{i} = (2,2)$ and $\boldsymbol{j} = (3,2)$: $\boldsymbol{d}(\boldsymbol{i}, \boldsymbol{j}) = (1,0)$, $\boldsymbol{D}(\boldsymbol{i}, \boldsymbol{j}) = (<, =)$