# COMP 515: Advanced Compilation for Vector and Parallel Processors

Prof. Vivek Sarkar
Department of Computer Science
Rice University
vsarkar@rice.edu

https://wiki.rice.edu/confluence/display/PARPROG/COMP515

# Worksheet 1 (Lecture 2)

Example:

```
      DO I = 1, N
         DO J = 1, M
            DO K = 2, L
S1             A(I+1, J, K-1) = A(I, J, K) + 10
            ENDDO
         ENDDO
      ENDDO
```

- Q: Are there any anti or output dependences in this example? If so, list them. If not, explain why not.
- A: No. There is only a flow dependence with distance vector (1, 0, -1)

# Dependence: Theory and Practice

**Allen and Kennedy, Chapter 2 (contd)**

# Distance Vectors (Summary)

- Consider a dependence in a loop nest of n loops
  - Statement $S_1$ with iteration vector i is the source of the dependence
  - Statement $S_2$ with iteration vector j is the sink of the dependence
  - n = number of common loops enclosing $S_1$ and $S_2$
- The distance vector, d(i,j), is a vector of length n such that:
  - $d(i,j)_k = j_k - i_k$
- In general, the distance vector is denoted as d, without the (i,j), because we conservatively assume that it applies to all pairs (i,j) that satisfy j = i + d
- We normalize distance vectors for loops in which the index step size is not equal to 1
  - It's usually simpler to convert all loops to have a step of +1 before computing distance vectors

# Direction Vectors (Summary)

- Definition 2.10 in the book:

  Suppose that there is a dependence from statement $S_1$ on iteration i of a loop nest of n loops and statement $S_2$ on iteration j, then the dependence direction vector is D(i,j) is defined as a vector of length n such that

$$D(i,j)_k = \begin{cases} \text{“<” if } i_k < j_k & \text{equivalently, if } d(i,j)_k > 0 \\ \text{“=” if } i_k = j_k & \text{equivalently, if } d(i,j)_k = 0 \\ \text{“>” if } i_k > j_k & \text{equivalently, if } d(i,j)_k < 0 \end{cases}$$

- A direction vector element summarizes a set of distances
  - “<” summarizes the set {1, 2, 3, …}
  - “=” summarizes the singleton set { 0 }
  - “>” summarizes the set {-1, -2, -3, …}
  - “*” denotes the union of “<”, “=”, and “>”
  - and so on …

# Implausible Distance & Direction Vectors

- A distance vector is implausible if its leftmost nonzero element is negative i.e., if the vector is lexicographically less than the zero vector
- Likewise, a direction vector is implausible if its leftmost non "=" component contains “>"
- By definition, no dependence in a sequential program can have an implausible distance or direction vector as this would imply that the sink of the dependence occurs before the source.

# Loop-carried and Loop-independent Dependences

- If in a loop statement $S_2$ depends on $S_1$, then there are two possible ways of this dependence occurring:

1. $S_1$ and $S_2$ execute on different iterations
   — This is called a loop-carried dependence.

2. $S_1$ and $S_2$ execute on the same iteration
   — This is called a loop-independent dependence.

- It is possible for both loop-carried and loop-independent dependences to occur between the same pair of statements

# Loop-independent dependences

- Definition 2.15. Statement $S_2$ has a loop-independent dependence on statement $S_1$ if and only if there exist two iteration vectors i and j such that:
  1) Statement $S_1$ refers to memory location M on iteration i, $S_2$ refers to M on iteration j, and i = j.
  2) There is a control flow path from $S_1$ to $S_2$ within the iteration.

Example:

```
DO I = 1, 10
S1      A(I) = ...
S2      ... = A(I)
ENDDO
```

# Loop-independent dependences

- No common loop is necessary. For instance:

```
DO I = 1, 10
S1      A(I) = ...
ENDDO
DO I = 1, 10
S2      ...  = A(20-I)
ENDDO
```

- An example with both loop-independent and loop-carried dependences:

```
DO I = 1, 9
S1       A(I) = ...
S2      ...  = A(10-I)
ENDDO
```

# Loop-carried dependence

- Definition 2.11
- Statement $S_2$ has a loop-carried dependence on statement $S_1$ if and only if $S_1$ references location M on iteration i, $S_2$ references M on iteration j and $d(i,j) > 0$ i.e., D(i,j) contains a "<" as leftmost non "=" component and is lexicographically positive.

Example:

```
DO I = 1, N
S1      A(I+1) = F(I)
S2      F(I+1) = A(I)
ENDDO
```

# Loop-carried dependence

- Level of a loop-carried dependence is the index of the leftmost non-“=” of D(i,j) for the dependence.

For instance:

```
DO I = 1, 10
   DO J = 1, 10
      DO K = 1, 10
S1       A(I, J, K+1) = A(I, J, K)
      ENDDO
   ENDDO
ENDDO
```

- Direction vector for S1 is (=, =, <)
- Level of the dependence is 3
- A level-k dependence between $S_1$ and $S_2$ is denoted by $S_1 \delta_k S_2$

# Homework 1 due in class on Sep 8th: Problems 2.2 and 2.3 from book

2.2 Construct all direction vectors for the following loop and indicate the type of dependence (true/anti/output) associated with each.

```
DO K = 1, 100
      DO J = 1, 100
            DO I = 1, 100
                  A(I+1, J, K) = A(I, J, 5) + B
            END DO
      END DO
END DO
```

2.3: Can any loop in Exercise 2.2 be parallelized? If so give a parallel version.