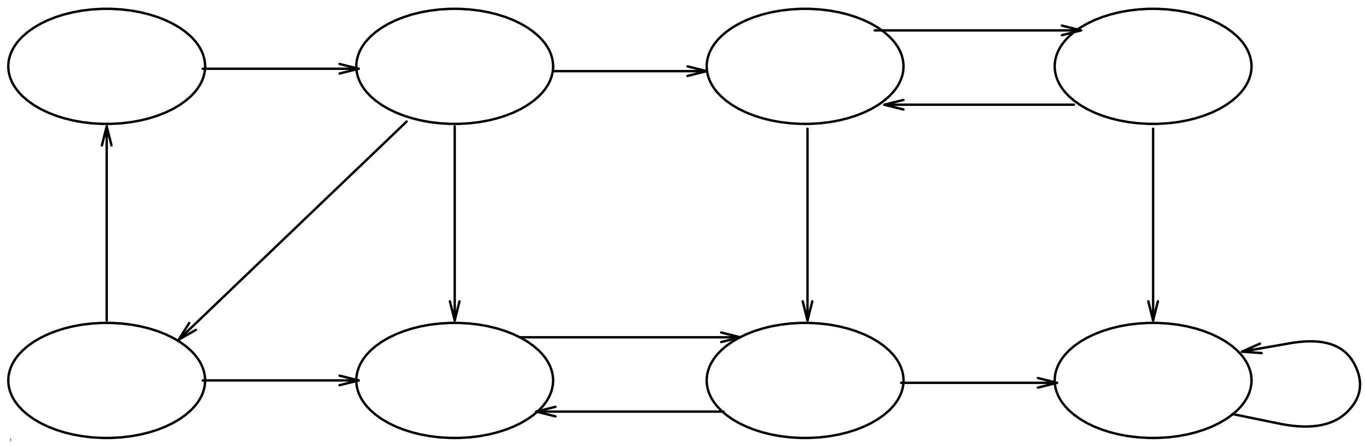Application to Parallelization and Vectorization (Section 2.3)

Phil Schielke

February 9, 1995

Overview

Present two algorithms that use dependence information to vectorize code.

Theorem

Fundamental Theorem of Loop Parallelization

The iterations of a given sequential loop may be executed in parallel if the loop carries no dependence.

Single statement loops that carry no dependence can be directly vectorized.

Question

"Can any statements in loops that carry dependences be directly vectorized?"

Answer

```
    DO I = 1, N
S1     A(I+1) = B(I) + C
S2     D(I) = A(I) + E
    ENDDO
```

This loop carries a dependence (S1 δ S2).

```
S1  A(2:N+1) = B(1:N) + C
S2  D(1:N) = A(1:N) + E
```

Loop Distribution

```
    DO I = 1, N
S1    A(I+1) = B(I) + C
    ENDDO
    DO I = 1, N
S2    D(I) = A(I) + E
    ENDDO
```

Theorem

Fundamental Theorem of Loop Vectorization

A statement in a loop can be vectorized by directly rewriting in FORTRAN 90 if the statement is not included in any cycle in the dependence graph.

Example

```
    DO I = 1, N
S1     A(I+1) = B(I) + C
S2     D(I) = A(I) + E(I)
S3     E(I+1) = D(I) + F
    ENDDO
```

Example

```
S1  A(2:N+1) = B(1:N) + C
    DO I = 1, N
S2     D(I) = A(I) + E(I)
S3     E(I+1) = D(I) + F
    ENDDO
```

Recall

Theorem 2.3: Any reordering transformation that does not alter loops 1 through k preserves any level-k dependence.

This suggests an algorithm!

Example

```
    DO I = 1, 100
S1      X(I) = Y(I) + 10
        DO J = 1, 100
S2          B(J) = A(J,N)
            DO K = 1, 100
S3              A(J+1,K) = B(J) + C(J,K)
            ENDDO
S4          Y(I+J) = A(J+1,N)
        ENDDO
    ENDDO
```

Example

```
DO I = 1, 100
   DO J = 1, 100
      B(J) = A(J,N)
      A(J+1:J+100) = B(J) + C(J,1:100)
   ENDDO
   Y(I+1:I+100) = A(2:101,N)
ENDDO
X(1:100) = Y(1:100) + 10
```

```
DO I = 1, 100
    codegen({S_2, S_3, S_4}, 2})
ENDDO
X(1:100) = Y(1:100) + 10
```

```
DO I = 1, 100
   DO J = 1, 100
      codegen({S_2, S_3}, 3})
   ENDDO
   Y(I+1:I+100) = A(2:101,N)
ENDDO
X(1:100) = Y(1:100) + 10
```

Summary

Introduced dependence – esp. *loop-carried* dependence.

Presented a vectorization algorithm using dependence information.

Next chapter: Constructing dependence graphs.

Digression