CS6013
Dependence Analysis

April 2, 2020

```
S1: a = b+c
S2: if a>10 goto L1
S3:    d = b*e
S4:    e = d + 1
S5: L1:d= e/2
```

Figure 1: Example program

If S1 preceed S2; we denote it as S1 $\triangleleft$ S2.

We say: S2 *depends* on S1 or S2 has a dependence on S1 or there is a dependence from S1 to S2. Types of dependence:

- Control dependence: S1 is a conditional check and depending on the value of the condition, S2 is executed or not. In Fig. 1, S3 control depends on S2. Denoted by S2 δ^c S3.
- Data dependence. S1 and S2: if S1 $\triangleleft$ S2. and
  - S1 computes/sets a value of a variable that S2 uses.
    flow dependence between S1 and S2. Also known as *true dependence*. Represented as S1 δ^f S2. In the above example: S3 δ^f S4.
  - S1 uses a value of a variable that is set by S2.
    Anti-dependence between S1 and S2. Represented as S1 δ^a S2. In the above example: S3 δ^a S4.
  - Both S1 and S2 write to the same variable.
    Output dependence between S1 and S2. Represented as S1 δ^o S2. In the above example: S3 δ^o S5.
  - Both S1 and S2 read the same variable.
    Input dependence between S1 and S2. Represented as S1 δ^i S2. In the above example: S3 δ^i S5 and S1 δ^i S3.

```
for i1 = 1 .. n1 {
    for i2 = 1 .. n2 {
    ...
        for ik = 1 .. nk {
            // all statements
        }
    }
}
```

Perfectly nested loops + loop index starts from 1, goes to some loop invariant value n, and increments only by 1 = Loop in canonical form.

The iteration space of the above loop nest - k-dimensional polyhedron. Each point in this k-dimensional space represents an iteration. Each iteration is a k-tuple.

Total iteration space: [1..n1] × [1..n2] × . . . [1..nk]

Each tuple in this iteration space, can be seen as index vectorwith k elements.

$\langle i1, i2, .., ik\rangle \prec \langle j1, j2, .., jk\rangle$

condition??

i_1, i_2

```
    for i1=1..3 {
        for i2=1 .. 4 {
S1        t = x + y
S2        a[i1,i2] = b[i1,i2] + c[i1,i2]
S3        b[i1,i2] = a[i1,i2-1] + d[i1+1,i2] + t
        }
    }
```

δ^f

✓ $S_2[i_1, i_2-1] \overset{\delta^f}{\triangleleft} S_3[i_1, i_2]$

$S_2[i_1, i_2] \underset{\delta^a}{\Delta} S_3[i_1, i_2]$