Polyhedral Operations

Logistics

–Intermediate reports late deadline is Friday March 30 at midnight

–HW6 (posted) and HW7 (posted) due April 5th

–Tuesday April 4th, help session during class with Manaf, Tomo, and Andy

–Distance students can send email to cs560@cs.colostate.edu with questions

Previously

- Misc and polyhedral representation

Today

- Representation of polyhedra with constraints
- Polyhedral operations using the constraint representation

Algorithms needed for automation

Operations on sets and relations

- Union iteration space sets
- Union relations that represent dependences
- Apply a relation to a set to model transforming a loop and to check transformation legality
- Compose two relations to model composing transformations

Scheduling

- Determine an efficient and legal schedule
- Determine which loops should be parallel

Storage Mapping

- If not using UOV, then need to do this in coordination with the scheduling

Code Generation

- Given a schedule and which loops to parallelize and/or tile, generate efficient code
- Code generation for parameterized tiles

Imperfectly Nested Loops (dependences I)

```
  /* Ring blur filter */
  for (i=1;i<length-1;i++)
    for (j=1;j<width-1;j++)
R     Ring[i][j]=(Img[i-1][j-1]+Img[i-1][j]+Img[i-1][j+1]+
                  Img[i][j+1]   +          Img[i][j-1]   +
                  Img[i+1][j-1]+Img[i+1][j]+Img[i+1][j+1])/8;

  /* Roberts edge detection filter */
  for (i=1;i<length-2;i++)
    for (j=2;j<width-1;j++)
P     Img[i][j]=abs(Ring[i][j]-Ring[i+1][j-1])+
                abs(Ring[i+1][j]-Ring[i][j-1]);
```

Figure 1. `Ring-Roberts` edge detection for noisy images

Representing computational sets as Polyhedra

Terminology referring to these sets

- Iteration space
- Domain
- Integer tuple space

Polyhedron

- A set $S \in \mathbb{R}^m$ is a polyhedron if there exists a system of finite inequalities $A\vec{x} \geq \vec{b}$ such that $P = \{\vec{x} \in \mathbb{R}^m | A\vec{x} \geq \vec{b}\}$
- Equivalently it is the intersection of finitely many half-spaces.
- Note that a polyhedron is a rational polyhedron. We have then been assuming intersection with the unit integer lattice.

Integral Polyhedron

- A set of the form: $P = \{\vec{x} \in \mathbb{Z}^m | Q\vec{x} \geq \vec{q}\}$
- Parametric family of polyhedra due to input parameters, or symbolic constants $P = \{\vec{x} \in \mathbb{Z}^m | Q\vec{x} \geq (\vec{q} + B\vec{p})\}$

Constraint Representation $P = \{\vec{x} \in \mathbb{Z}^m | Q\vec{x} \geq (\vec{q} + B\vec{p})\}$

Implementation

- Store a coefficient matrix for all the constraints
- Associate each column with an iterator or a parameter/symbolic constant

Interpretation of the above representation (assume all column vectors)

$$P = \{\begin{pmatrix} \vec{x} \\ \vec{p} \end{pmatrix} \in \mathbb{Z}^m | \begin{bmatrix} Q & B & \vec{q} \end{bmatrix} \begin{pmatrix} \vec{x} \\ \vec{p} \\ 1 \end{pmatrix} \geq \vec{0}\}$$

- "Parameterized family of polyhedra is just a single higher-dimensional polyhedron." Foundations II notes by Sanjay Rajopadhye.

Example

```
for (i=0; i<N; i++) {
  for (j=i+1; j<N; j++) {
  A[j][i] = A[j][i]/A[i][i];
  for (k=i+1; k<N; k++) {
    A[j][k] -= A[i][k]*A[j][i];
}}}
```


Operation: Intersection

Intersection between polyhedral sets

- When you intersect two polyhedral sets the result is a polyhedral set.

$$P = \{\vec{x} \in \mathbb{Z}^m | \vec{x} \in P_1 \wedge \vec{x} \in P_2\}$$

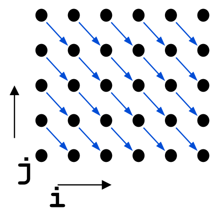
$$P_1 = \{\begin{pmatrix} \vec{x} \\ \vec{p_1} \end{pmatrix} \in \mathbb{Z}^m | \begin{bmatrix} Q_1 & B_1 & \vec{q_1} \end{bmatrix} \begin{pmatrix} \vec{x} \\ \vec{p_1} \\ 1 \end{pmatrix} \geq \vec{0}\}$$

$$P_2 = \{\begin{pmatrix} \vec{x} \\ \vec{p_2} \end{pmatrix} \in \mathbb{Z}^m | \begin{bmatrix} Q_2 & B_2 & \vec{q_2} \end{bmatrix} \begin{pmatrix} \vec{x} \\ \vec{p_2} \\ 1 \end{pmatrix} \geq \vec{0}\}$$

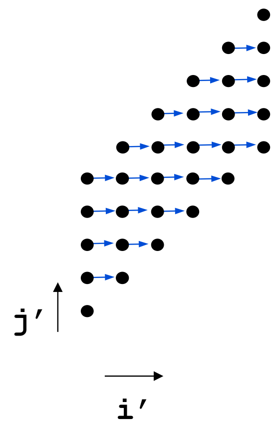
What operations do we need?
(Review applying transformations)

Original code

```
do i = 1,6
  do j = 1,5
     A(i,j) = A(i-1,j+1)+1
  enddo
enddo
```



Distance vector: (1, -1)



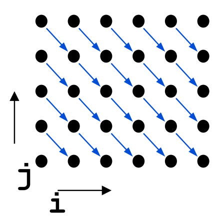
Skewing:

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} = \begin{bmatrix} i \\ i+j \end{bmatrix}$$

Transforming the Dependences and Array Accesses

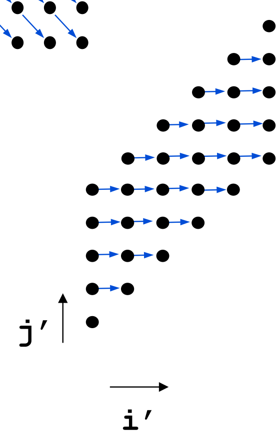
Original code

```
do i = 1,6
  do j = 1,5
     A(i,j) = A(i-1,j+1)+1
  enddo
enddo
```



Dependence vector:

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



New Array Accesses:

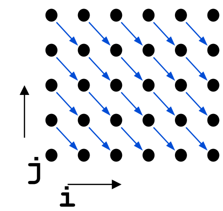
$$A\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}\right) = A(i,j)$$

$$A\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} i' \\ j' \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}\right) = A(i', j' - i')$$

$$A\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} + \begin{bmatrix} -1 \\ 1 \end{bmatrix}\right) = A(i-1, j+1)$$

$$A\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} i' \\ j' \end{bmatrix} + \begin{bmatrix} -1 \\ 1 \end{bmatrix}\right) = A(i'-1, j' - i' + 1)$$

Transforming the Loop Bounds



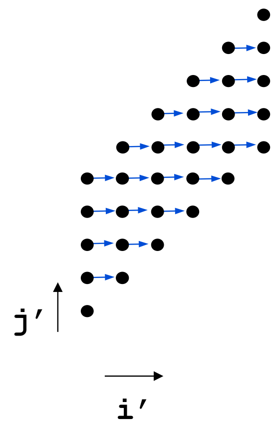
Original code

```
do i = 1,6
    do j = 1,5
        A(i,j) = A(i-1,j+1)+1
    enddo
enddo
```

Bounds:

$$\begin{bmatrix} -1 & 0 \\ 1 & 0 \\ 0 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} \leq \begin{bmatrix} -1 \\ 6 \\ -1 \\ 5 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 0 \\ 1 & 0 \\ 0 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} i' \\ j' \end{bmatrix} \leq \begin{bmatrix} -1 \\ 6 \\ -1 \\ 5 \end{bmatrix}$$



Transformed code

```
do i' = 1,6
    do j' = 1+i',5+i'
        A(i',j'-i') = A(i'-1,j'-i'+1)+1
    enddo
enddo
```


Specifying Loop Transformations

Unimodular

$$f(\vec{i}) = T\vec{i}$$

Polyhedral/Affine Transformations (also called a schedule or Change of Basis)

$$f(\vec{i}) = T\vec{i} + \vec{k}$$

Kelly and Pugh (Presburger in general case)

$$\{\vec{i} \rightarrow \vec{x} \mid \vec{x} = T\vec{i} + \vec{k}\}$$

Remaining notes are all written

Next Time

Lecture

– Yet again more operations on polyhedral sets and relations

Schedule

– Project intermediate report late submission is March 20th
– April 3rd will be a lab day during class, Manaf, Tomo, and Andy will help people with AlphaZ and Pluto. Distance students can email questions or use the discussion board.
– HW6 and HW7 will BOTH be due April 4th

~~Slide 4~~

Slide 5

3/29/12

$$\boxed{\begin{array}{c} 0 \le i < N \\ i+1 \le \boxed{j < N} \end{array}}$$

$$P = \left\{ \begin{pmatrix} i \\ j \\ N \end{pmatrix} \in \mathbb{Z}^3 \;\middle|\; \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & -1 \\ -1 & 1 & 0 & -1 \\ 0 & -1 & 1 & -1 \end{bmatrix} \begin{pmatrix} i \\ j \\ N \\ 1 \end{pmatrix} \ge \vec{0} \right\}$$

$-i > -N$

$-i-1 \ge -N$

$-i-1+N \ge 0$

$j-i-1 \ge 0$

$j < N$

$-j > -N$

$-j-1+N \ge 0$

Slide 6

$$P = \left\{ \begin{pmatrix} \vec{x} \\ \vec{p}_1 \\ \vec{p}_2 \end{pmatrix} \in \mathbb{Z}^{|x|+|p_1|+|p_2|} \;\middle|\; \begin{bmatrix} Q_1 & B_1 & 0 & \vec{q}_1 \\ Q_2 & 0 & B_2 & \vec{q}_2 \end{bmatrix} \begin{pmatrix} \vec{x} \\ \vec{p}_1 \\ \vec{p}_2 \\ 1 \end{pmatrix} \ge \vec{0} \right\}$$

$P = P_1 \cap P_2$

CS 560 $\boxed{3/29/12}$

affine transformation

$T_{I \to I'} = \{ i \to x \mid x = Bi + k$

k is a constant vector

foundations II a transformation is a change of basis

Apply transformation to an Iterationspace

$$I = \left\{ \begin{pmatrix} i \\ p \end{pmatrix} \middle| \; A \begin{pmatrix} i \\ p \\ 1 \end{pmatrix} \geq 0 \right\}$$

$$I' = T_{I \to I'}(I)$$

definitions

$s = r1(s2)$ r1 ≡ relation
s2 = set
s = set

$y \in s$ iff $\exists x$ st $x \to y \in r1$ AND $x \in s2$

Transform Array Accesses

array access

$$F_{I \to D} = \{ i \to d \mid d = Ai + k \}$$

$$= \left\{ i \to d \mid d = \begin{bmatrix} A & k \end{bmatrix} \begin{pmatrix} i \\ 1 \end{pmatrix} \right\}$$

new array access function

$$F_{I' \to D} = \{ T_{I \to I'}(i) \to F_{I \to D}(i) \mid i \in I \}$$

$$= F_{I \to D} \circ T^{-1}_{I \to I'} \quad \text{let } T_{I' \to I} = T^{-1}_{I \to I'}$$

$$= F_{I \to D} \circ T_{I' \to I}$$

Transform the data dep relation

$$D_{I\to I} = \{ i \to j \mid A\begin{pmatrix} i \\ j \\ p \\ 1 \end{pmatrix} \geq 0 \}$$

$$B\begin{pmatrix} i \\ j \\ p \\ 1 \end{pmatrix} = 0$$

new data dep relation

$$D_{I'\to I'} = T_{I\to I'} \circ D_{I\to I} \circ T_{I'\to I}$$

Implenting Inverse

Let $r = \{ x \to y \mid y = Bx + c \}$

$y \to x \in r^{-1}$ iff $x \to y$ in r

$$r^{-1} = \{ y \to x \mid y = Bx + c \}$$
$$\mid x = B^{-1}(y - c) \}$$

NOTE: B needs to admit left inverse

Implementing compose

Let $r1 = \{ y \to z \mid z = Ay + k \}$

$r2 = \{ x \to y \mid y = Bx + c \}$

$r = r1 \circ r2$

def $x \to z \in r$ iff $\exists y$ st $x \to y \in r2 \wedge$
$y \to z \in r1$

$r = r1 \circ r2$

$= \{x \to z \mid \exists y : x \to y \in r2 \land y \to z \in r1\}$

$= \{x \to z \mid \exists y : y = Bx + c \land z = Ay + k\}$

$= \{x \to z \mid \underset{-z}{z} = ABx + \underset{-z}{Ac} + k\}$

$$\mid \begin{bmatrix} AB & A & -\vec{I} & k \end{bmatrix} \begin{bmatrix} x \\ c \\ z \\ 1 \end{bmatrix}$$

Apply function to a set

Let $r1 = \{x \to y \mid y = Bx + k\}$

$s2 = \{x \mid Qx \geq (q + Bp)\}$

$s = r1(s2)$

def: $y \in s$ iff $\exists x$ st $\underline{x \to y \in r1} \land x \in s2$

$s = \{y \mid \exists x : y = Bx + k \land Qx \geq (q + Bp)\}$

$\{y \mid QB^{-1}(y - k) \geq (q + Bp)\}$