Automating Scheduling

Logistics

–HW8 due tomorrow, April 11^{th}

Previously

– Polyhedral operations using the constraint representation

Today

– Automating scheduling with the Farkas lemma

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

Affine Scheduling

A schedule maps each iteration to a virtual time

$$\theta(\vec{i}) = T \begin{pmatrix} \vec{i} \\ \vec{p} \\ 1 \end{pmatrix}$$

– The number of rows in T is the dimensionality of the schedule.

– The number of rows in T is also the number of outermost sequential loops.

Citation: http://www.cse.ohio-state.edu/~pouchet/lectures/doc/888.11.3.pdf

Scheduling in the Polyhedral Model

Legality

– The schedule must respect all the dependences.

– Let's turn dependence relations into constraints on the schedule solution set.

  – If iteration $\vec{i_R}$ of statement R needs to execute before iteration $\vec{i_S}$ of statement S, then the schedules for statement R and S need to satisfy the following constraint:

$$\theta_R(\vec{i_R}) \prec \theta_S(\vec{i_S})$$

One-dimensional schedules

$$\theta_R(\vec{i_R}) < \theta_S(\vec{i_S})$$

Constraint for schedule legality

Time delta

- between statement instances with dependences,
- needs to be non-negative over the dependence polyhedron

$$\Delta_{R,S} = \theta_S(\vec{i_S}) - \theta_R(\vec{i_R}) - 1 >= 0$$

<Example dependence polyhedron done on paper>

Turning this observation into scheduling constraints

Affine form of Farkas lemma

- Let D be a nonempty polyhedron defined by $A\vec{i} + \vec{b} \geq 0$.
- Any affine function f(i) is non-negative everywhere in D if and only if it is a positive affine combination of the constraints for D:

$$f(\vec{i}) = \lambda_0 + \vec{\lambda}^T(A\vec{i} + \vec{b})$$

$$\text{with } \lambda_0 \geq 0 \text{ and } \vec{\lambda} \geq \vec{0}$$

where λ_0 and $\vec{\lambda}^T$ are called the Farkas multipliers.

Citation: Thies et al 2007, A step towards unifying schedule and storage optimization

Building intuition about the Farkas lemma

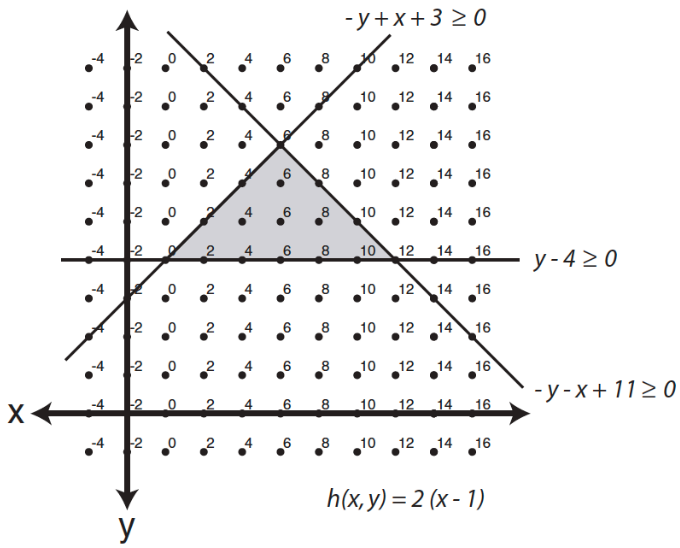


Fig. 2. An illustration of Farkas' lemma. The affine form $h(x, y) = 2 \cdot (x - 1)$ is nonnegative within the shaded polyhedron. Thus, it can be expressed as a nonnegative affine combination of the faces of that polyhedron: $h(x, y) = 2 \cdot (-y + x + 3) + 2 \cdot (y - 4)$.

Using the Farkas lemma

Assume the following dependence polyhedron

$$D_{R \to S} = \{[\vec{i} \to \vec{j} \mid A \begin{bmatrix} \vec{i} \\ \vec{j} \\ \vec{p} \\ 1 \end{bmatrix} \geq \vec{0} \text{ and } B \begin{bmatrix} \vec{i} \\ \vec{j} \\ \vec{p} \\ 1 \end{bmatrix} = \vec{0}\}$$

Assume a schedule function of the form

$$\theta_R(\vec{i}) = \vec{v}^T \vec{i} + \vec{b}$$
$$\theta_S(\vec{j}) = \vec{w}^T \vec{j} + \vec{c}$$

We need $\Delta_{R,S} = \theta_S(\vec{i}) - \theta_R(\vec{j}) - 1 >= 0$

The process of determining set of legal schedules

(1) Change all of the equality constraints in $D_{R \to S}$ to inequality constraints.

$$D_{R \to S} = \{[\vec{i} \to \vec{j} \mid A' \begin{bmatrix} \vec{i} \\ \vec{j} \\ \vec{p} \\ 1 \end{bmatrix} \geq \vec{0}\}$$

(2) Use the Farkas lemma to create a set of constraints for the schedule.

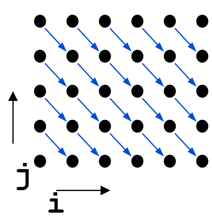
$$\begin{aligned} \theta_S(\vec{i}) - \theta_R(\vec{j}) - 1 \quad &= \quad \lambda_0 + \vec{\lambda}^T (A' \begin{bmatrix} \vec{i} \\ \vec{j} \\ \vec{p} \\ 1 \end{bmatrix}) \\ \lambda_0 \geq 0 \quad &\text{and} \quad \vec{\lambda} \geq \vec{0} \\ \theta_R(\vec{i}) \quad &= \quad \vec{v}^T \vec{i} + \vec{b} \\ \theta_S(\vec{j}) \quad &= \quad \vec{w}^T \vec{j} + \vec{c} \end{aligned}$$

(3) Solve for v, w, b, and c vector constraints by projecting out lambdas.

Example of using the Farkas lemma

Original code

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



(1) Dependence polyhedron

(2) Farkas lemma to set up constraints

(3) Project out lambdas to determine set of legal schedules

Next Time

Lecture

- How do we select a best schedule?
- Other approaches for automating the scheduling problem.

Schedule

- HW8 due tomorrow, April 11th

(slide 5)

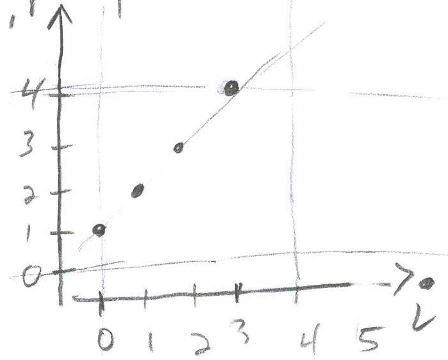
```
for (i=0; i<5; i++) {
    A[i] = A[i-1] + sin(i);
}
```

Dep relation

$$\{ [i] \rightarrow [i'] \mid 0 \le i < 5 \ \wedge \ 0 \le i' < 5 \ \wedge \ i = i'-1 \ \wedge \ i < i' \ \}$$

$A[i]$ $\quad$ $A[i'-1]$

Dep. polyhedron



initial schedule

$\theta(i) = i$

R & S statement are the same

need $\Delta_{R,S} = \theta(i') - \theta(i) - 1 \ge 0$

$= i' - i - 1 \ge 0$

$= (i+1) - i - 1 \ge 0$

$0 \ge 0$ ✓

$\theta(i) = i + 1$

$\theta(i) = i + k$ $\quad$ legal schedules

any constant

(slide 7) $\boxed{\text{General}}$

$$D = \{ \vec{i} \mid A\vec{i} + \vec{b} \geq 0 \}$$

$$f(\vec{i}) = \lambda_0 + \vec{\lambda}^T (A\vec{i} + \vec{b})$$

$$\boxed{\begin{array}{l} \lambda_0 \geq 0 \\ \vec{\lambda} \geq \vec{0} \end{array}}$$

$\boxed{\text{Specific Example}}$

$$D = \{ [x, y] \mid -y + x + 3 \geq 0 \wedge \\ y - 4 \geq 0 \wedge \\ -y - x + 11 \geq 0 \}$$

$$f([x, y]) = 2(x-1)$$

$$\boxed{2(x-1) = \lambda_0 + \lambda_1(-y + x + 3) + \lambda_2(y-4) + \lambda_3(-y - x + 11)}$$

$$0 = (\lambda_0 + 2 + 3\lambda_1 - 4\lambda_2 + 11\lambda_3) + (\lambda_1 - 2 - \lambda_3)x + (-\lambda_1 + \lambda_2 - \lambda_3)y$$

$$\lambda_0 + 2 + 3\lambda_1 - 4\lambda_2 + 11\lambda_3 = 0$$

$$\lambda_1 - 2 - \lambda_3 = 0 \Rightarrow \lambda_1 = 2 + \lambda_3$$

$$-\lambda_1 + \lambda_2 - \lambda_3 = 0 \Rightarrow \lambda_2 = \lambda_1 + \lambda_3 = 2 + 2\lambda_3$$

4/10/12 ③

(1a) Dependence polyhedron for distance vector $(1,-1)$

$$D = \{ [i,j] \rightarrow [i',j'] \mid \begin{array}{l} i = i'-1 \\ j = j'+1 \\ 1 \le i, i' \le 6 \\ 1 \le j, j' \le 5 \} \end{array}$$

(1b) Change all equality constraints to inequality constraints

$$D = \{ [i,j] \rightarrow [i',j'] \mid \begin{array}{ll} i - i' + 1 \ge 0 & r1 \\ -i + i' - 1 \ge 0 & \\ j - j' + 1 \ge 0 & r3 \\ -j + j' - 1 \ge 0 & \\ i - 1 \ge 0 & \\ i' - 1 \ge 0 & \\ 6 - i' \ge 0 & \\ 6 - i \ge 0 & \\ \vdots & \end{array} \qquad \overbrace{\begin{bmatrix} 1 & 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & -1 & 1 \\ 0 & -1 & 0 & 1 & -1 \\ & & \vdots & & \end{bmatrix}}^{A'} \begin{pmatrix} i \\ j \\ i' \\ j' \\ 1 \end{pmatrix}$$

(2) Use Farkas lemma to create a set of constraints for the schedule

$$\Theta([i',j']) - \Theta([i,j]) - 1 = \lambda_0 + \vec{\lambda} A' \begin{pmatrix} i \\ j \\ i' \\ j' \\ 1 \end{pmatrix}$$

$$\boxed{\text{Let } \Theta([x,y]) = v_0 x + v_1 y + b_0}$$

$$v_0 i' + v_1 j' + b_0 - v_0 i - v_1 j - b_0 - 1 = \lambda_0 + \vec{\lambda}^T A' \begin{pmatrix} i \\ j \\ i' \\ j' \\ 1 \end{pmatrix}$$

r1: $v_0 i' + v_1 j' - v_0 i - v_1 j - 1 = \lambda_0 + \lambda_1 i - \lambda_3 i' + \lambda_5$

$\vdots$

r3: $v_0 i' + v_1 j' - v_0 i - v_1 j - 1 = \lambda_0 + \lambda_2 j - \lambda_4 j' + \lambda_5$