Module 17: Loops

# Lecture 34: Symbolic Analysis

## The Lecture Contains:

- Symbolic Analysis
- Example:
- Triangular Lower Limits
- Multiple Loop Limits
- Exit in The Middle of a Loop
- Dependence System Solvers
- Single Equation
- Simple Test
- GCD Test
- Extreme Value Test

Previous Next

- Lower bound of each unknown is zero
- Trip count for outer loop is 99
  Therefore, upper bound of $i_1^d, i_1^u,$, are 98
- Trip count for inner loop is $1 + i_1$

  Therefore upper bound for inner loop is

$$i_2^d \le i_1^d \qquad i_2^u \le i_1^u$$

$$\begin{pmatrix} -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} i_1^d \\ i_1^u \\ i_2^d \\ i_2^u \end{pmatrix} \le \begin{pmatrix} 0 \\ 98 \\ 0 \\ 98 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$-i_1^d \le 0 \quad i_1^d \le 98$$
$$-i_1^u \le 0 \quad i_1^u \le 98$$
$$-i_2^d \le 0 \quad -i_1^d + i_2^d \le 0 \; or \; i_2^d \le i_1^d$$
$$-i_2^u \le 0 \quad -i_1^u + i_2^u \le 0 \; or \; i_2^u \le i_1^u$$

### Symbolic Analysis

- User variables may occur in subscript expressions
- Treat each user variable as another unknown
- If coefficient of user variable is zero, it is eliminated

Previous Next

Example:

for i = 1 to n-1
for j = 1 to n-i+1
B[i,j] = B[i,n-1]
endfor
endfor

$$1 + i_1^d = 1 + i_2^u$$
$$1 + i_2^d = 1 - i_1^u + n$$

Triangular Lower Limits

- Compiler may use semi-normalized space
- Same iteration space shape must be used
- Existence of integer solution is the same
- Dependence distance/direction can be different

for i = 2 to n
for j = i+1 to n+i+1
B[i,,j] = B[i-1,,j-1] + C[i]
endfor
endfor

The normalized statement is:

B[2 + $i_1$, 3 + $i_1$ + $i_2$] = B[$i_1$ + 1, $i_1$ + $i_2$ + 2]+ · · ·

Previous Next

Module 17: Loops

Lecture 34: Symbolic Analysis

The dependence equations are:

$$2 + i_1^d = 1 + i_1^u$$
$$3 + i_1^d\, i_2^d = 2 + i_1^u + i_2^u$$

and the constraints are:

$$0 \le i_1^d \le n-2$$
$$0 \le i_1^u \le n-2$$
$$0 \le i_2^d \le n-1$$
$$0 \le i_2^u \le n-1$$

- Original loop limits for the inner loop are linear in the outer loop
- Retain shape of the iteration space using semi-normalized loops
- Use new iteration variable for the inner loop

$$j_2 = i_2 + i_1$$

The dependence equations are:

$$2 + i_1^d = 1 + i_1^u$$
$$3 + j_2^d = 2 + j_2^u$$

and the constraints are:

$$0 \le i_1^d \le n-2$$
$$0 \le i_1^u \le n-2$$
$$i_1^d \le j_2^d \le i_1^d + n - 1$$
$$i_1^u \le j_2^u \le i_1^u + n - 1$$

Previous Next

## Multiple Loop Limits

- Lower limit is the maximum of several expressions
- Upper limit is the minimum of several expressions

```
for i = 1 to 8
for j = max(i-3,1), min(i,5)
A[i+1,,j+1] = A[i,,j] + B[i,,j]
endfor
endfor
```

## Normalization gives:

for $0 \leq i_1 \leq 7$
for $max(i_1 - 3,0) \leq i_2 \, min(i_1, 4)$
$A[i_1 + 2, i_2 + 2] = A[i_1 + 1, i_2 + 1] + \cdots$
end for
endfor
with dependence equations: $i_1^d + 1 = i_1^u$ and $i_2^d + 1 = i_2^u$

## Exit in The Middle of a Loop

Some statements may execute more number of times than others

1. j = 0
2. loop
3. j = j + 1
4. A[j] =···
5. if j > 10 then exit
6. = A[j+1]
7. endloop

line number 4 executes eleven times, therefore,

$$i_d + 1 = i_u + 2$$
$$0 \leq i_d \leq 11$$
$$0 \leq i_u \leq 10$$

Module 17: Loops

Lecture 34: Symbolic Analysis

Dependence System Solvers

- Integer programming problem
- Exact solvers very expensive (exponential or worse)
- Dependence systems trade-off between
  - Efficiency, speed of solver
  - Precision, reducing, number of *'false positives'*
- All systems are conservative
  - Never return *'no soln'* when there is a solution
  - May return *'possible soln'* where there is no solution
- Three possible results from a solver
  - *'no soln'* means there is no integer solution
  - *'has soln'* means there is an integer solution. Some solvers may enumerate solution
  - *'possible soln'* means result is inexact.
    Solver cannot prove that there is no solution or a soln
- Characteristics of solvers:
  - Cost
  - Applicability
  - Imprecision

## Single Equation

A single dependence eqn can be written as:

$$\sum_{k=0}^{n} a_k i_k = c$$

Where n: Number of unknowns
$a_k$: coefficients

Previous Next

## Simple Test

Simplest test when a single loop (2 unknowns) and a single dimension with linear subscript expression.

***Definition*** $a_1 i^d + c_1$

***Use*** $a_2 i^u + c_2$

And assume same coefficient $(a_1 = a_2 = a)$

$$a i^d - a i^u = c_2 - c_1$$

- This has integer solution if gcd of coefficients divides rhs
- In this case *gcd(a, a)* = *a*

  Therefore, if a divides $C_2 - C_1$ then there is a dependence

## Example

For I = 2 to 10 do

$A[2*I+2] = A[2*I-2] + B[I]$

Endfor

Using normalized loop,

$$A[6+2i] = A[2+2i] + B[2+i]$$

Therefore, $6 + 2i^d = 2 + 2i^u$

$$2i^d - 2i^u = -4$$

Therefore, $i^d - i^u = -2$

- Now determine actual dependence solution
  - Either a flow dependence with distance $d^f$ Therefore, $i^d + d^f = i^u$
  - Or, anti-dependence with distance $d^a$ Therefore, $i^u + d^a = i^d$

    Therefore, $d^f = +2$ or $d^a = -2$

Previous Next

Module 17: Loops

Lecture 34: Symbolic Analysis

### GCD Test

- There is an integer solution to the dependence eqn. when $gcd\ a_1, a_2, a_3, \ldots, a_n$ divides c
- If not, then there are no integer solutions regardless of the bounds.
- Applies to Single Dependence eqn.
- Inexpensive test; finding gcd is very efficient

### Extreme Value Test

- Find the extreme values of the expression in dependence eqn.
  $\sum_{k=0}^{n} a_k i_k = c$
  The region of $R^n$ is bounded by loop limits and other constraints
- Method finds lower and upper bounds of the function
- Value of c must lie between lower and upper bounds
- Efficient but inexact test
- Does not enforce restriction to integer soln.

### Example

If M>0 then
For I=1 to 10 do
$A[I] = A[I+M] + B[I]$
Endfor
Endif
$i^d + 1 = i^u + M + 1$
Therefore, $-M + i^d - i^u = 0$
The constraints are:

$$0 \leq i^d \leq 9$$
$$0 \leq i^u \leq 9$$

Modify $1 \leq M \longrightarrow 1 \leq M \leq +\infty$

Previous Next

| Step | Lower Bound | Upper Bound |
|---|---|---|
| original eqn | $-M + i^d - i^u$ | $-M + i^d - i^u$ |
| eliminate $i^u$ | $-M + i^d - 9$ | $-M + i^d$ |
| eliminate $i^d$ | $-M - 9$ | $-M + 9$ |
| eliminate M | $-\infty$ | 8 |

Since 0 lies in between $-\infty$ and 8, extreme value test assumes there is a dependency.

- To determine kind and direction of dependence, apply direction vector constraint
- First apply $i^d < i^u$constraint.
  In integer domain , this becomes $i^d \leq i^u - 1$
  Also, $0 \leq i^d < i^u \leq 9$
  Therefore,

$$1 \leq M \leq \infty$$
$$0 \leq i^d \leq \begin{cases} 8 \\ i^u - 1 \end{cases}$$
$$\left.\begin{matrix} 1 \\ i^d + 1 \end{matrix}\right\} \leq i^u \leq 9$$

- Extreme value method can use only one bound
- Only one of the upper bounds of $i^d$ and lower bound of $i^u$ can be used

Previous Next

| Step | Lower Bound | Upper Bound |
|---|---|---|
| original eqn | $-M + i^d - i^u$ | $-M + i^d - i^u$ |
| eliminate $i^u$ | $-M + i^d - 9$ | $-M + i^d - (i^d + 1) = -M - 1$ |
| eliminate $i^d$ | $-M - 9$ | $-M - 1$ |
| eliminate M | $-\infty$ | $-2$ |

No dependence when $i^d < i^u$

$For\ i^d > i^u$

$$1 \leq M \leq \infty$$
$$1 \leq i^d \leq 9$$
$$0 \leq i^u \leq i^d - 1$$

| Step | Lower Bound | Upper Bound |
|---|---|---|
| original eqn | $-M + i^d - i^u$ | $-M + i^d - i^u$ |
| eliminate $i^u$ | $-M + i^d - i^d + 1$ | $-M + i^d$ |
| eliminate $i^d$ | $-M + 1$ | $-M + 9$ |
| eliminate M | $-\infty$ | 8 |

Therefore, there is a dependence

$For\ i^d = i^u$

$$-M + i^d - i^u = -M + i^d - i^d = -M$$

Extreme values of M are $\infty$ and 1

Therefore, dependence with $i^d \mid= i^u$ cannot exist

Previous Next