

# Loop Iterator Live-out Value

Louis-Noël Pouchet

pouchet@cse.ohio-state.edu

Dept. of Computer Science and Engineering, the Ohio State University

September 2010

**888.11**



# Objectives

- ▶ Solve a “real” compiler problem
- ▶ Write understandable algorithms (communication skill!)
- ▶ Prove the correctness of an algorithm
- ▶ Study the complexity of an algorithm

**Problem: computation of the exit value of loop iterators**

# The Problem

Compute the exit value of all loop iterators of a given input C program

## Example (Input program)

```
for (i = 1; i < 3; ++i)
  for (j = 1; j < 3; ++j)
    S(i, j)
```

## Example (Solution)

```
i = 3;
j = 3;
```

# Step-by-step methodology

- 1 **Problem definition:** Understand and define the problem
- 2 **Examples:** Find various example, and compute the desired output by hand
- 3 **Restriction:** Find an algorithm, maybe restricted to simpler cases
- 4 **Generalization:** Generalize the algorithm to work on all cases
- 5 **Proof:** Prove the algorithm is complete and correct
- 6 **Complexity:** Study the complexity of the algorithm

## An Instance of the Problem

**Problem:** the exit value of loop iterators is not preserved when doing affine transformations

### Example (Input code)

```
for (i = 1; i < 3; ++i)
  for (j = 1; j < 3; ++j)
    S(i, j);
printf("%d", i); // prints '3'
```

### Example (After shifting by -1)

```
for (i = 0; i < 2; ++i)
  for (j = 0; j < 3; ++j)
    S1(i + 1, j);
printf("%d", i); // prints '2'
```

## When Does This Problem Occur?

- ▶ When we perform transformations on the input code that changes the loop structure, or
- ▶ When we rename loop iterators
- ▶ **AND** when a loop iterator value is read after the transformed block execution

**We want our transformation framework to generate correct code!**

## Context of Application: Affine Framework

**Must work on any sub-program that can be represented/transformed in the affine framework:**

- ▶ loop bounds are affine expressions of surrounding loop iterators and parameters
- ▶ same for conditionals
- ▶ we do not care about the actual statement body: no impact
  - ▶ property: if the program is a scop, then a loop iterator is not modified except by the loop increment statement
- ▶ loops iterator increase by step of 1 after 1 execution of the body, until the exit test is false

### Example

```
for (i = lbi; i < Ubi; ++i)
  for (j = lbj(i); j < Ubj(i); ++j)
    for (k = lbk(i, j); k < Ubk(i, j); ++k)
      if (expr(i, j, k))
        S1(i, j, k)
```

Here,  $lb$ ,  $Ub$  and  $expr$  are affine expressions of global parameters and their arguments

# Input and Output of the algorithm

Input:

- ▶ The AST of the sub-program
  
- ▶ We also have at hand the affine representation of the sub-program

Output:

- ▶ An AST to be inserted after the sub-program
  
- ▶ This AST assigns to each loop iterator their exit value in the original program



# Methodology

Before designing the algorithm:

- ▶ Are we clear on the input? On the output?
- ▶ Do we need to learn something about the chosen input/output?
  - ▶ AST representation
  - ▶ Input language (eg, specifics of `for` loops in C)
- ▶ Can we manually write a few input/output examples?
- ▶ Do we need additional restrictions on the input?

## Examples of Input and Output [1/9]

### Example (Input program)

```
for (i = 1; i < 3; ++i)
  for (j = 1; j < 3; ++j)
    S(i, j);
```

### Example (Desired output)

```
i = 3;
j = 3;
```

## Examples of Input and Output [2/9]

### Example (Input program)

```
for (i = 1; i < N; ++i)
  S(i, j);
```

**Exercise: write the output**

## Examples of Input and Output [2/9]

### Example (Input program)

```
for (i = 1; i < N; ++i)
  S(i, j);
```

**Exercise: write the output**

### Example (Desired output)

```
if (1 >= N)
  i = 1;
else
  i = N;
```

## Examples of Input and Output [3/9]

### Example (Input program)

```
for (i = 1; i < N; ++i)
  for (j = 0; j < M; ++j)
    S(i, j);
```

**Exercise: write the output**

## Examples of Input and Output [3/9]

### Example (Input program)

```
for (i = 1; i < N; ++i)
  for (j = 0; j < M; ++j)
    S(i, j);
```

**Exercise: write the output**

### Example (Desired output)

```
if (1 >= N)
  i = 1;
else {
  i = N;
  if (0 >= M)
    j = 0;
  else
    j = M;
}
```

## Examples of Input and Output [4/9]

### Example (Input program)

```
for (i = 1; i < N; ++i) {  
  for (j = 0; j < M; ++j)  
    S(i, j);  
  for (j = 0; j < P; ++j)  
    S(i, j);  
}
```

**Exercise: write the output**

## Examples of Input and Output [4/9]

### Example (Input program)

```
for (i = 1; i < N; ++i) {  
  for (j = 0; j < M; ++j)  
    S(i, j);  
  for (j = 0; j < P; ++j)  
    S(i, j);  
}
```

**Exercise: write the output**

### Example (Desired output)

```
if (1 >= N)  
  i = 1;  
else {  
  i = N;  
  if (0 >= M)  
    j = 0;  
  else  
    j = M;  
  if (0 >= P)  
    j = 0;  
  else  
    j = P;  
}
```



## Examples of Input and Output [5/9]

### Example (Input program)

```
for (i = 1; i < N; ++i) {  
  for (j = 0; j < M; ++j)  
    S(i, j);  
  for (j = 0; j < min(N, P); ++j)  
    S(i, j);  
}
```

**Exercise: write the output**

## Examples of Input and Output [5/9]

### Example (Input program)

```
for (i = 1; i < N; ++i) {  
  for (j = 0; j < M; ++j)  
    S(i, j);  
  for (j = 0; j < min(N, P); ++j)  
    S(i, j);  
}
```

**Exercise: write the output**

### Example (Desired output)

```
if (1 >= N)  
  i = 1;  
else {  
  i = N;  
  if (0 >= M)  
    j = 0;  
  else  
    j = M;  
  if (N > P) {  
    if (0 >= P)  
      j = 0;  
    else  
      j = P;  
  }  
  else {  
    if (0 >= N)  
      j = 0;  
    else  
      j = N;  
  }  
}
```

## Examples of Input and Output [6/9]

### Example (Input program)

```
for (i = 1; i < N; ++i) {  
  for (j = 0; j < M; ++j)  
    S(i, j);  
  if (N > M)  
    for (j = 0; j < min(N, P); ++j)  
      S(i, j);  
}
```

**Exercise: write the output**

## Examples of Input and Output [6/9]

### Example (Input program)

```
for (i = 1; i < N; ++i) {  
  for (j = 0; j < M; ++j)  
    S(i, j);  
  if (N > M)  
    for (j = 0; j < min(N, P); ++j)  
      S(i, j);  
}
```

**Exercise: write the output**

### Example (Desired output)

```
if (1 >= N)  
  i = 1;  
else {  
  i = N;  
  if (0 >= M)  
    j = 0;  
  else  
    j = M;  
  if (N > M) {  
    if (N > P) {  
      if (0 >= P)  
        j = 0;  
      else  
        j = P;  
    }  
  }  
  else {  
    if (0 >= N)  
      j = 0;  
    else  
      j = N;  
  }  
}
```

## Examples of Input and Output [7/9]

### Example (Input program)

```
for (i = 1; i < N; ++i) {  
  for (j = i; j < M; ++j)  
    S(i, j);  
  if (i > M)  
    for (j = i; j < min(N, P); ++j)  
      S(i, j);  
}
```

**Exercise: write the output**

## Examples of Input and Output [7/9]

### Example (Input program)

```
for (i = 1; i < N; ++i) {
  for (j = i; j < M; ++j)
    S(i, j);
  if (i > M)
    for (j = i; j < min(N, P); ++j)
      S(i, j);
}
```

**Exercise: write the output**

### Example (Desired output)

```
if (1 >= N)
  i = 1;
else {
  i = N;
  if (N - 1 >= M)
    j = N - 1;
  else
    j = M;
  if (N > M) {
    if (N > P) {
      if (N - 1 >= P)
        j = N - 1;
      else
        j = P;
    }
  }
  else {
    if (N - 1 >= N)
      j = N - 1;
    else
      j = N;
  }
}
```

## Examples of Input and Output [8/9]

### Example (Input program)

```
for (i = 1; i < N; ++i)
  for (j = max(N, M); j < P; ++j)
    S(i, j);
```

**Exercise: write the output**

## Examples of Input and Output [8/9]

### Example (Input program)

```
for (i = 1; i < N; ++i)
  for (j = max(N, M); j < P; ++j)
    S(i, j);
```

**Exercise: write the output**

### Example (Desired output)

```
if (1 >= N)
  i = 1;
else {
  i = N;
  if (N >= M) {
    if (N < P)
      j = P;
    else
      j = N;
  }
  else {
    if (M < P)
      j = P;
    else
      j = M;
  }
}
```



## Examples of Input and Output [9/9]

### Example (Input program)

```
for (i = 1; i < N; ++i)
  for (j = max(i, M); j < P; ++j)
    S(i, j);
```

**Exercise: write the output**

## Examples of Input and Output [9/9]

### Example (Input program)

```
for (i = 1; i < N; ++i)
  for (j = max(i, M); j < P; ++j)
    S(i, j);
```

**Exercise: write the output**

### Example (Desired output)

```
if (1 >= N)
  i = 1;
else {
  i = N;
  if (N - 1 >= M) {
    if (N < P)
      j = P;
    else
      j = N - 1;
  }
  else {
    if (M < P)
      j = P;
    else
      j = M;
  }
}
```

# Examples of Input and Output

**Exercise: write sample input and the corresponding output**