Static Loop Bound Analysis of C Programs Based on Flow Analysis and Abstract Interpretation

M. de Michiel, *A. Bonenfant*, H. Cassé and P. Sainrat

RTCSA 2008

Plan

Context

Worst-Case Execution Time for Critical Real-Time Systems

- Flow Fact Analysis
  - Path Retrieval
  - Loop Bound Analysis
- Timing Analysis
  - Global Effect Analysis (cache, branch prediction)
  - Local Effect Analysis (pipeline)
- WCET computation

Loop Bound Analysis: Our Tool

Input/Output

- INPUT: A C program
- OUTPUT : *total* and *max* expressions of loop bounds

Example total/max

```
int boucle (int n){
   ...
   for (i=0;i<n;i++) ...
   ...
}

void main...{
   boucle(5);
   boucle(10);
}
```

total = 10 and max = 5

Method

Method: 3 steps

- context insensitive identification/normalisation of increment and loop bound expression,
- context sensitive construction of *total* and *max* expressions,
- propagation of context to compute final *total* and *max*

Techniques

- Abstract Interpretation
- Extension of Ammarguellat[SIGPLAN'90] method

Step 2 Algorithm

Step 2 Algorithm

```
treat_internal_loop (Bi,Bj,Bk,Bl,totalBl,maxBl,
    ContextBi):
  replace Bk by totalBl and maxBl in Bj
  evaluate the context of Bj until fix point
  construct new totalBl (sum or product) and maxBl
      (max or unchanged)
  if i=j
    totalBl = totalBl in ContextBi
    maxBl = maxBl in ContextBi
  else
    treat_internal_loop (Bi,Bj-1,Bk-1,Bl,totalBl,
        maxBl,ContextBi)
```

Example

C program

```
n = 10;
for(i = 0; i < n; i++)
{
   for(j = i+1; j <= n; j++)
   {
      w = i+1+j;
      for(k = 0; k <= i; k+=2)
         w = i+1+k;
      a = w;
   }
}
```

Example

Step 1 : Identification/Normalisation

- increment variables i, j and k and increment value 1, 1 and 2
- construction of expB1 = $\lfloor$(n-i)/1$\rfloor$, expB2 = $\lfloor$(n-j)/1$\rfloor$+1 and expB3 = $\lfloor$(i-k)/2$\rfloor$+1
- normalisation using varBi and increment +1

```
n = 10;
i = 0;
loop(B1,for,varB1,condB1=(i<n),expB1,{
    j = 0
    loop(B2,for,varB2,condB2=(j<=n),expB2,{
      w = i+1+j;
      k = 0;
      loop(B3,for,varB3,condB3=(k<=i),expB3,{
        w = i+1+k;
        k+=2;
      })
      a = w;
      j++;
    })
    i++;
  })
```

Example

Step 2 : Construction of total & max expressions: B1

treat_internal_loop B1 B1 B1 B1 totalB1 maxB1 $\sigma_{\text{before-loopB1}}$

$\text{totalB1} = \text{maxB1} = \text{expB1} = \lfloor (\text{n-i})/1 \rfloor$

$\sigma_{\text{before-loopB1}} = [\text{n} \mapsto 10\,;\, \text{i} \mapsto 0]$

$\text{totalB1} = \text{maxB1} = [\![\lfloor (\text{n-i})/1 \rfloor]\!] \sigma_{\text{before-loopB1}} = \lfloor (10 - 0)/1 \rfloor$

Example (2)

Step2 : Construction of total & max expressions: B2

treat_internal_loop B1 B1 B2 B2 totalB2 maxB2 $\sigma_{\text{before-loopB1}}$

$$\text{totalB2} = \text{maxB2} = \text{expB2} = \lfloor(\text{n-j})/1\rfloor+1$$

```
B1 with B2 replaced = B1-2:
loop(B1, for, varB1, condB1, {
  j = i+1;
  totalB2 = maxB2 = n-j+1;
  i++;
})
```

$$\sigma_{B1-2} = \begin{bmatrix} \text{j} \mapsto \text{i+varB1+1}; \\ \text{totalB2} \mapsto \text{n-(i+varB1+1)+1}; \\ \text{maxB2} \mapsto \text{n-(i+varB1+1)+1}; \\ \text{i} \mapsto \text{i+varB1+1} \end{bmatrix}$$

$$\text{totalB2} = \sum_{\text{varB1}=0}^{\text{totalB1}-1} 10 - (0 + \text{varB1} + 1) + 1$$

$$\text{maxB2} = max_{\text{varB1}=0}^{\text{totalB1}-1} 10 - (0 + \text{varB1} + 1) + 1$$

Example (3-1)

Step2 : Construction of total & max expressions: B3

treat_internal_loop B1 B2 B3 B3 totalB3 maxB3 $\sigma_{\text{before-loopB1}}$

totalB3 = maxB3 = expB3 = $\lfloor$(i-k)/2$\rfloor$+1

- at B2 level

```
B2 with B3 replaced: B2-3
loop(B2, for, varB2, condB2, {
  w = i+1+j;
  k = 0;
  totalB3=maxB3=floor((i-k)/2)+1;
  a = w;
  j++;
})
```

$$\sigma_{B2-3} = \begin{bmatrix} w \mapsto i+1+j+varB2; \\ k \mapsto 0; \\ totalB3 \mapsto \lfloor i/2 \rfloor+1; \\ maxB3 \mapsto \lfloor i/2 \rfloor+1; \\ a \mapsto i+1+j+varB2; \\ j \mapsto j+varB2+1 \end{bmatrix}$$

$$totalB3 = maxB3 \times totalB2 = (\lfloor i/2 \rfloor+1) \times (n-j+1)$$

$$maxB3 = \lfloor i/2 \rfloor+1$$

Example (3-2)

Step2 : Construction of total & max expressions: B3

- at B1 level

  treat_internal_loop B1 B1 B2 B3 totalB3 maxB3 $\sigma_{\text{before-loopB1}}$

```
B1 with B2 replaced = B1-3:
loop(B1, for, varB1, condB1,
  {
    j = i+1;
    totalB3=(floor(i/2)+1)*(n-j+1);
    maxB3 = floor(i/2)+1;
    i++;
  })
```

$$\sigma_{B1-3} = \begin{bmatrix} j \mapsto i+varB1+1 \\ totalB3 \mapsto (\lfloor (i+varB1)/2 \rfloor + 1) \times (n - (\lfloor (i+varB1)/2 \rfloor + 1) + 1) \\ maxB3 \mapsto \lfloor (i+varB1)/2 \rfloor + 1; \\ i \mapsto i+varB1+1 \end{bmatrix}$$

$$totalB3 = \sum_{varB1=0}^{totalB1-1} (\lfloor (i+varB1)/2 \rfloor + 1) \times (n - (\lfloor (i+varB1)/2 \rfloor + 1) + 1)$$

$$maxB3 = max_{varB1=0}^{totalB1-1} \lfloor (i + varB1)/2 \rfloor + 1$$

Example (3-3)

Step 2 : Construction of total & max expressions: B3

$$[\![\ \text{totalB3}]\!]\sigma_{\text{before-loopB1}} = \sum_{\text{varB1}=0}^{\text{totalB1}-1} (\lfloor(0+\text{varB1})/2\rfloor + 1) \times (10 - (\lfloor(0+\text{varB1})/2\rfloor + 1) + 1)$$

$$[\![\text{maxB3}]\!]\sigma_{\text{before-loopB1}} = max_{\text{varB1}=0}^{\text{totalB1}-1} \lfloor(0+\text{varB1})/2\rfloor + 1$$

Example (4)

Step2 : Construction of total & max expressions: propagation of context

- $[\![\text{totalB1}]\!]\sigma_{\text{before-loopB1}} = [\![\text{maxB1}]\!]\sigma_{\text{before-loopB1}} = \lfloor(10-0)/1\rfloor = 10$
- $[\![\text{totalB2}]\!]\sigma_{\text{before-loopB1}} = \sum_{\text{varB1}=0}^{9} 10-(0+\text{varB1}+1)+1$
- $[\![\ \text{maxB2}]\!]\sigma_{\text{before-loopB1}} = max_{\text{varB1}=0}^{9} 10-\text{varB1} = 10$
- $[\![\ \text{totalB3}]\!]\sigma_{\text{before-loopB1}} = \sum_{\text{varB1}=0}^{9} (\lfloor(0+\text{varB1})/2\rfloor+1)\times(10-(\lfloor(0+\text{varB1})/2\rfloor+1)+1)$
- $[\![\text{maxB3}]\!]\sigma_{\text{before-loopB1}} = max_{\text{varB1}=0}^{9} \lfloor(0+\text{varB1})/2\rfloor+1$

Example (5)

Step 3: total and max computation

- totalB1= maxB1 = 10
- totalB2=$\sum_{varB1=0}^{9} 10 - varB1 = 55$

  maxB2=$max_{varB1=0}^{9} 10 - varB1 = 10$
- totalB3= $\sum_{\text{varB1}=0}^{9} (\lfloor \text{varB1}/2 \rfloor + 1) \times (10 - (\lfloor \text{varB1}/2 \rfloor + 1) + 1)$
  $= NOCOMP$

  maxB3=$max_{varB1=0}^{9} \lfloor varB1/2 \rfloor + 1 = 5$

  totalB3 $<= 5 \times 55 = 275$

Example (6)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<flowfacts>
  <function name="main" executed="true">
    <loop loopId="1" executed="true" line="6" source="rtcsa.c"
                    totalcount="10" maxcount="10" exact="true">
      <loop loopId="2" executed="true" line="10" source="rtcsa.c"
                      totalcount="55" maxcount="10" exact="true">
        <loop loopId="3" executed="true" line="16" source="rtcsa.c"
                        totalcount="275" maxcount="5" exact="false">
        </loop>
      </loop>
    </loop>
  </function>
</flowfacts>
```

Mälardalen WCET Benchmark

Results

- Mälardalen WCET Benchmark
- Ermedahl[WCET'07]
-

Total	P	BLT	Ratio	Exact	T (sec)
Our analysis	33	138	84%	total: 121 max: 114	45.26
Ermedahl	28	104	63%	84	499.25
Total	35	164			

Mälardalen WCET Benchmark

Comparison

- 50% of additional loops bounded in: **duff, fft1, lcdnum, ns, qurt**
- Ermedahl better results on
  - **fac:** recursive, not treated
  - **fir:** multiple increment, less bounds found
  - **ndes:** loop in `if` statements, bounds over-estimated
  - **ud:** `break`, bounds over-estimated
- Additional programs we analyse: **compress, lms, minver, sqrt, st**

Current & Further Work

Current Work

- Normalisation Extention: multiple increment, indirect increment...
- "Single" Recursivity (by rewriting)
- Arrays and Pointers

Further Work

- Extend total and max Formulae Resolution
- "Complex" Recursivity (Fibonacci type, mutual recursivity...)
- Interval Analysis
- Complex Control Flow (`break`)
- ...

THANK YOU!