To appear in the proceedings of the IEEE IWIA 2002 Workshop

Tight Non-Linear Loop Timing Estimation

Robert A. van Engelen* & Kyle A. Gallivan†

Department of Computer Science
and School of Computational Science and Information Technology
Florida State University
Tallahassee, FL 32306-4530
{engelen,gallivan}@cs.fsu.edu

Abstract

Parametric worst case execution time (WCET) bounds are useful in removing restrictions, such as known loop bounds, on algorithms for important applications such as scheduling for real-time embedded systems. However, current parametric approaches have difficulties with multiple loop nests that include non-rectangular loops, zero-trip loops, and/or loops with non-unit strides. These difficulties may result in the increased looseness of the upper bound, increased complexity in determining the bound, or loss of the upper bound property of the estimate. In this paper, we propose a new framework to compute tight parameterized WCET bounds for multiple loop nests that may include non-rectangular loops, zero-trip loops, and loops with non-unit strides. The framework requires only very simple symbolic manipulation capabilities and is restricted only by certain monotonic properties required of the iteration spaces of the inner loops.

1. Introduction

Worst-case execution time (WCET) estimation has important applications in scheduling and real-time embedded systems. Traditional timing analysis statically determines the maximum execution time of a program, which requires the maximum number of loop iterations to be statically known. Recent work on WCET estimation [4] incorporated work on scheduling [5] to bound the number of iterations of non-rectangular loops to derive tight WCET estimations. The method requires the bounds of the outer loop to be constant or to be bounded by constants, resulting in a loosening of the WCET bound.

Parametric timing analysis produces formulas, where the unknown values affecting the execution are parameterized [8]. This formula can be quickly evaluated at run-time with real-time values to support dynamic scheduling decisions. However, current parametric approaches have difficulties with multiple loop nests that include non-rectangular loops, zero-trip loops, and/or loops with non-unit strides. These difficulties may result in the increased looseness of the upper bound, increased complexity in determining the bound, or loss of the upper bound property of the estimate.

In this paper, we propose a new framework to compute tight parameterized WCET bounds for multiple loop nests that may include non-rectangular loops, zero-trip loops, and loops with non-unit strides. The proposed framework requires only very simple symbolic manipulation capabilities on representations of polynomials. The technique is restricted only by certain monotonic properties required of the iteration spaces of the inner loops.

The remaining part of this paper is organized as follows. In Section 2 we present the proposed framework for tight loop timing estimation. Section 3 illustrates the application of the technique to an example loop nest and discusses the results. Finally, some concluding remarks are given in Section 4.

2. Tight Loop Timing Estimation

In this section, we present the proposed framework for tight loop timing estimation.

2.1. General Formulation

Consider the following generalized form of a loop:

```
for I = a to b do
    stmt
od
```

∗ Supported in part by NSF grants CCR-0105422 and CCR-9904943
† Supported in part by NSF grants CCR-0105422 and EIA-0072043

where I denotes the loop counter variable with lower bound a and upper bound b. The entire loop body is denoted by *stmt*, which may consist of a block of statements that may include inner loops.

To analyze the timing of the loop, a normalized iteration variable i will be used to denote the i^{th} loop iteration with $i = 0, \ldots, b-a$. Furthermore, let $time(stmt, i)$ denote the execution time of the loop body *stmt* in the i^{th} iteration and let $time(hdr, i)$ denote the execution time of the loop header in the i^{th} iteration. The total execution time ω of this loop can be expressed as an accumulation of the execution times of individual operations in the loop. This is depicted by the loop fragment

```
ω := time(hdr)
for i = 0 to b − a do
    ω += time(stmt, i) + time(hdr, i)
od
```

Obviously, the value of ω at the termination of the loop holds the exact total execution time of the loop fragment. This formulation provides insight into the mechanics of the proposed generalized worst-case execution time (WCET) analysis framework and to show that it subsumes conventional static WCET analysis techniques.

The proposed framework allows the use of non-constant upper bounds on the execution time of the loop header and body, e.g.

$$\begin{aligned} time(hdr) &\leq c_0 \\ time(stmt, i) + time(hdr, i) &\leq T(i) \end{aligned}$$

for all $i = 0, \ldots, b-a$. The use of a constant upper bound $T(i) = c_1$ on the loop body execution time results in a conventional form of WCET analysis. The use of $T(i) = 1$ results in solving the loop iteration count problem, e.g. for scheduling. Given sufficient information on the behavior of the program, the timing analysis can be made more accurate when a non-constant upper bound $T(i)$ is provided on the loop body execution time, such that

$$time(stmt, i) + time(hdr, i) \leq T(i) \leq c_1$$

for all $i = 0, \ldots, b-a$.

For example, $T(i)$ may be a linear function in i in case of a triangular loop nest, where the inner loop iteration space depends on the outer loop counter variable:

```
for i = a to b do
    for j = 0 to i do
        stmt
    od
od
```

An inner loop may also result from a library call such as a string and/or memory move operation.

The proposed framework handles these cases with a unified approach. For any choice of bounds $c_0 \geq 0$ and $T(i) \geq 0$, we can assert that the total time ω of a loop is bounded by

$$\begin{aligned} \omega &= time(hdr) + \sum_{i=0}^{b-a} time(stmt, i) + time(hdr, i) \\ &\leq c_0 + \sum_{i=0}^{b-a} T(i) \quad . \end{aligned}$$

Due to the semantics of the $\sum$ operator, the bound is valid only if the loop has a non-empty iteration space, i.e. $a < b$. Otherwise, when $a \geq b$ the loop has an empty iteration space and is called a *zero-trip loop*.

To count the number of loop iterations in the presence of potential zero-trip loops, Haghighat [3] proposed a method that exploits summations over special functions that are similar to the Dirac delta operator. Sakellariou [5] proposed the use of a summation operator with semantics that differ from the usual semantics of the summation operator $\sum$, that satisfies

$$\sum_{i=a}^{b} X(i) = -\sum_{i=b}^{a} X(i) \quad .$$

For example, the computation of the number of iterations of the loop with iteration space $i = 0, \ldots, 3$ and $j = i, \ldots, 2$ by summing gives

$$\sum_{i=0}^{3} \sum_{j=i}^{2} 1 = \sum_{i=1}^{3} i - 1 = 2$$

while the actual count is 6. The bounds derived by summation are not upper bounds on the size of iteration space and cannot be used in WCET estimation.

In this text, we will use the $\widetilde{\sum}$ operator to denote Sakellariou's summation operator, which is defined by

$$\widetilde{\sum_{i=a}^{b}} X_i = \begin{cases} \sum_{i=a}^{b} X_i & \text{if } a \leq b \\ 0 & \end{cases} \quad .$$

With this summation operator, we can derive a bound that is valid in the presence of potential zero-trip loops

$$\begin{aligned} \omega &= time(hdr) + \widetilde{\sum_{i=0}^{b-a}} time(stmt, i) + time(hdr, i) \\ &\leq c_0 + \widetilde{\sum_{i=0}^{b-a}} T(i) \quad . \end{aligned}$$

Unfortunately, the computation of the WCET with $\widetilde{\sum}$ is problematic for multi-dimensional loop nests with symbolic bounds and strides. We present a novel solution based on

Newton's forward formula for the interpolating polynomial. If the upper bound $T(i)$ on the execution time of the loop body is polynomial of finite order k, then it follows, from the Newton's forward formula for the interpolating polynomial, that there exist constant coefficients ϕ_j, $j = 0, \ldots, k$, such that

$$T(i) = \sum_{j=0}^{k} \phi_j \binom{i}{j} \quad . \tag{1}$$

Observe that

$$\sum_{i=0}^{n-1} T(i) = \sum_{j=0}^{k} \phi_j \binom{n}{j+1} \tag{2}$$

for all $n > 0$, because

$$\sum_{i=0}^{n-1} \sum_{j=0}^{k} \phi_j \binom{i}{j} = \sum_{j=0}^{k} \phi_j \sum_{i=0}^{n-1} \binom{i}{j} = \sum_{j=0}^{k} \phi_j \binom{n}{j+1} \quad .$$

As a result, a potentially problematic loop iteration count summation with symbolic bounds can be translated into a simpler summation with constant bounds. The summation on the right-hand side of Equation (2) can be computed with symbolic bound n, provided that the coefficients ϕ_j for $j = 0, \ldots, k$ can be obtained from the polynomial $T(i)$.

To effectively deal with zero-trip loops, we define the polynomial

$$p(\Phi, n) = \sum_{j=0}^{k} \phi_j \binom{n}{j+1} \quad . \tag{3}$$

It can be easily verified that

$$p(\Phi, \max(0, n)) = \sum_{i=0}^{n-1} T(i)$$

which follows immediately from Equation (2) for $n > 0$, and by noting that both sides reduce to 0 for $n \leq 0$. This also implies that $p(\Phi, n)$ is monotonically increasing with increasing size $n > 0$ of the iteration space of the loop because $T(i) \geq 0$ for all $i = 0, \ldots, n$. In addition, note that

$$p(0) = \sum_{j=0}^{k} \phi_j \binom{0}{j+1} = 0 \quad .$$

Hence,

$$p(\Phi, \max(0, n)) = \max(0, p(\Phi, n)) \quad .$$

Based on this, the WCET is defined by

$$WCET = \max(c_0, c_0 + p(\Phi, b - a + 1)) \tag{4}$$

which bounds the real execution time ω of the loop

$$\omega \leq c_0 + \sum_{i=0}^{b-a} T(i) = \max(c_0, c_0 + p(\Phi, b - a + 1)) \quad .$$

When the loop bounds a and b are symbolic, the result is a parameterized WCET estimation.

It follows that conventional WCET analysis is a simple application of Definition (4) with $k = 0$ and $\phi_0 = T(i) = c_1$. This amounts to the computation of WCET by

$$\begin{aligned} WCET &= \max(c_0, c_0 + p(c_1, b - a + 1)) \\ &= \max(c_0, c_0 + c_1(b - a + 1)) \end{aligned}$$

where c_1 is the upper bound on the execution time of the loop body

$$time(stmt, i) + time(hdr, i) \leq c_1 \quad .$$

The bound c_1 may be established by an iterative algorithm that simulates the cache behavior of the loop body repeatedly until the timing of the loop body converges, see e.g. [4, 8]. This WCET bound can be very loose for triangular loop nests, because the inner loop is bounded by a non-negative constant over the entire iteration space of the inner loop.

The definition of the WCET bound (4) requires the coefficients ϕ_j of Φ to be computed in order to evaluate (3). The coefficients can be obtained with the CR algebra [1, 9] by application of CR rewrite rules on a symbolic form of polynomial $T(i)$. For example, suppose that the execution time of the loop body is bounded by polynomial $T(i) = c_1 + c_2\, i + c_3\, i^2$. This execution time bound can be the result of an inner loop nest that is triangular. An example of such a loop nest can be found in the TRFD code, which has been extensively studied for compiler optimization [2, 3, 6]. Application of the CR rewrite rules on $T(i)$ gives

$$\begin{aligned} & c_1 + c_2\, i + c_3\, i^2 \stackrel{CR}{\Rightarrow} \\ &= c_1 + c_2 \{0, +, 1\}_i + c_3 \{0, +, 1\}_i^2 \\ &= c_1 + c_2 \{0, +, 1\}_i + c_3 \{0, +, 1, +, 2\}_i \\ &= c_1 + \{0, +, c_2\}_i + \{0, +, c_3, +, 2\, c_3\}_i \\ &= \{c_1, +, c_2 + c_3, +, 2\, c_3\}_i \quad . \end{aligned}$$

See also [7] for a list of CR algebra rewrite rules. Hence, the CR coefficients are $\phi_0 = c_1$, $\phi_1 = c_2 + c_3$, and $\phi_2 = 2\, c_3$. The closed form expression that defines the WCET of the loop over the possibly empty iteration space size $b - a + 1$ is obtained by using Definition (4)

$$\begin{aligned} & WCET \\ &= \max(c_0, c_0 + p(\Phi, b - a + 1)) \\ &= \max(c_0, c_0 + (c_1 - \tfrac{1}{2}c_2 + \tfrac{1}{6}c_3)(b - a + 1) \\ &\quad + \tfrac{1}{2}(c_2 - c_3)(b - a + 1)^2 + \tfrac{1}{3}c_3(b - a + 1)^3) \quad . \end{aligned}$$

The CR algebra derives the coefficients ϕ_j in $\mathcal{O}(k^3)$ time by application of rewrite rules on a symbolic form of $T(i)$. However, the coefficients can be determined in $\mathcal{O}(k^2)$ time by computing

$$\begin{pmatrix}\phi_0\\\phi_1\\\phi_2\end{pmatrix} = \begin{pmatrix}1&0&0\\0&1&1\\0&0&2\end{pmatrix}\begin{pmatrix}c_1\\c_2\\c_3\end{pmatrix}$$

where the matrix is based on an underlying Pascal's triangle. This matrix has fixed coefficients that only depend on the polynomial order of $T(i)$.

Instead of applying Definition (3) to compute $p(\Phi, n)$, the linear system

$$\begin{pmatrix}0\\\phi_0\\\phi_1\\\phi_2\end{pmatrix} = \begin{pmatrix}1&0&0&0\\0&1&1&1\\0&0&2&6\\0&0&0&6\end{pmatrix}\begin{pmatrix}p_0\\p_1\\p_2\\p_3\end{pmatrix}$$

can be solved to determine the coefficients of polynomial $p(\Phi, n) = p_0 + p_1\, n + p_2\, n^2 + p_3\, n^3$. Solving this system, e.g. with backward substitution, gives

$$\begin{aligned} p_0 &= 0\\ p_1 &= c_1 - \tfrac{1}{2}c_2 + \tfrac{1}{6}c_3\\ p_2 &= \tfrac{1}{2}(c_2 - c_3)\\ p_3 &= \tfrac{1}{3}c_3 \end{aligned}$$

which can then be directly used to compute the WCET with Definition (4).

Note that the WCET can also be computed with

$$WCET = c_0 + \sum_{i=0}^{b-a} c_1 + c_2\, i + c_3\, i^2 \quad .$$

However, straight forward computation of this summation requires more extensive symbolic manipulation. Furthermore, the approach in general suffers from complications with non-linear and symbolic bounds, zero-trip loops, and loop nests with strides. In contrast, the use of the WCET metric given by Definition (4) avoids these complications and is faster to compute provided that the coefficients ϕ_j can be efficiently obtained from any polynomial $T(i)$.

2.2. Timing of Loops With Strides

The timing of a loop with iteration bounds a and b and a non-unit stride s can also be estimated with this framework. If this loop is non-zero-trip, i.e. $(b-a)/s \geq 0$, then

$$p(\Phi, \lfloor \tfrac{b-a}{s} \rfloor + 1) \leq p(\Phi, \tfrac{b-a}{s} + 1) \quad .$$

This relation is valid because p is monotonically increasing. Note that the relation is an equality when s evenly divides $b - a$. The WCET of a possibly zero-trip loop with strides is defined by

$$WCET = \max(c_0, c_0 + p(\Phi, \tfrac{b-a}{s} + 1)) \qquad (5)$$

which bounds the real execution time ω of the loop, where a, b, and $s \neq 0$ are arbitrary (symbolic) expressions, and Φ is the CR representation of the polynomial WCET of the loop body statement. If the remainder $(b-a) \bmod s$ is small, the bound is tight.

2.3. Timing of Loop Nests

Consider the general representation of a double loop nest

```
for I = L_I to U_I step S_I do
    for J = L_J(I) to U_J(I) step S_J do
        stmt
    od
od
```

which has the normalized iteration space

$$\begin{aligned} i &= 0, \ldots, \lfloor \tfrac{U_I - L_I}{S_I} \rfloor\\ j &= 0, \ldots, \lfloor \tfrac{U_J(L_I + S_I\, i) - L_J(L_I + S_I\, i)}{S_J} \rfloor \quad . \end{aligned}$$

Suppose the following execution time bounds are given

$$\begin{aligned} time(outer\_loop\_hdr) &\leq c_0\\ time(inner\_loop\_hdr) &\leq c_1\\ time(stmt) + time(inner\_loop\_hdr) &\leq T(i,j) \quad . \end{aligned}$$

The WCET of the inner loop is computed first. Let Ψ denote the CR representation of the polynomial of $T(i,j)$ with respect to j (i can be considered constant, since it is loop-invariant in the J loop). Then the WCET of the inner loop by Definition (5) is

$$WCET = \max(c_1, c_1 + p(\Psi, \tfrac{U_J(L_I + S_I\, i) - L_J(L_I + S_I\, i))}{S_J} + 1)$$

which bounds the real execution time ω of the inner loop. To derive the WCET of the entire loop nest, the WCET expression of the inner loop is required to be polynomial in i. This requires the elimination of the $\max$ operation from the WCET expression above. The $\max$ operation can be eliminated if it can be shown that

$$p(\Psi, \tfrac{U_J(i + L_I) - L_J(i + L_I))}{S_J} + 1) \geq 0 \quad .$$

This is verified as follows. If the bounds on the inner loop $L_J(I)$ and $U_J(I)$ are polynomial expressions in I, then

$$q(i) = p(\Psi, \tfrac{U_J(i + L_I) - L_J(i + L_I))}{S_J} + 1)$$

is guaranteed to be polynomial in i. By Equation (1), there exist CR coefficients Φ for polynomial $q(i)$. To conclude that $q(i) \geq 0$ so that the $\max$ operation can be eliminated, we verify whether $\phi_j \geq 0$ for $j = 0, \ldots, k$. This holds if the size of the iteration space of the inner loop monotonically increases through the iterations of the outer loop.

The $\max$ operation can still be eliminated when the size of the iteration space of the inner loop monotonically decreases through the iterations of the outer loop. To do so, the iteration order of the inner loop is reversed and the j parameter of WCET bound function $T(i, j)$ on the execution time of the inner loop is changed accordingly.

Finally, the $\max$ operation can be eliminated when dealing with inner loops whose iteration space is not monotonic. In that case, conservative bounds on the size of the iteration space of the inner loop are used. That is, (symbolic) constant bounds L and U are used such that $L \leq L_J(L_I + S_I\, i)$ and $U \geq U_J(L_I + S_I\, i)$. This results in $q(i)$ being equal to a constant $\phi_0 \geq 0$ if $L \leq U$. However, this change to the inner loop bounds results in a loosening of the WCET bound of the entire loop nest.

Finally, the WCET of the entire loop nest is computed by application of Definition (5)

$$WCET = \max(c_0, c_0 + p(\Phi, \tfrac{U_I - L_I}{S_I} + 1))$$

which bounds the real execution time $\omega \leq WCET$ of the entire loop nest.

The proposed framework for WCET estimation is applicable to multi-dimensional loop nests. The algorithm iterates the steps described above by starting with the estimation of the WCETs of the inner most loops and by working toward the outer loops until the WCET of the entire loop nest is computed.

3. An Example

In this section, we present the application of the proposed framework to an example loop nest. We compare the WCET to other parameterized WCET estimation techniques and discuss the results.

Consider the loop nest

```
for I = 1 to N do
    for J = I to I*I-2 step 2 do
        stmt
    od
od
```

Suppose the following execution time bounds are given

$$\begin{aligned} time(outer\_loop\_hdr) &\leq c_0 \\ time(inner\_loop\_hdr) &\leq c_1 \\ time(stmt) + time(inner\_loop\_hdr) &\leq c_2 \quad . \end{aligned}$$

Normalization of the loop nest yields the iteration space

$$\begin{aligned} i &= 0, \ldots, N-1 \\ j &= 0, \ldots, \lfloor \tfrac{i+i^2}{2} \rfloor - 1 \quad . \end{aligned}$$

Note that the inner loop is zero-trip for $i = 0$. The WCET of the inner loop is

$$WCET = \max(c_1, c_1 + p(\Psi, \tfrac{i+i^2}{2}))$$

where Ψ has a single CR coefficient $\psi_0 = c_2$. Hence,

$$c_1 + p(\Psi, \tfrac{i+i^2}{2}) = c_1 + \tfrac{1}{2} c_2 (i + i^2) \quad .$$

The CR coefficients of this polynomial are

$$\begin{pmatrix} \phi_0 \\ \phi_1 \\ \phi_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} c_1 \\ \frac{1}{2} c_2 \\ \frac{1}{2} c_2 \end{pmatrix}$$

which gives $\phi_0 = c_1$, $\phi_1 = c_2$, and $\phi_2 = c_2$. All of the coefficients of are non-negative, because $c_1 \geq 0$ and $c_2 \geq 0$. Therefore, the $\max$ can be eliminated from the WCET estimation of the inner loop

$$WCET = c_1 + p(\Psi, \tfrac{i+i^2}{2}) \quad .$$

To compute the WCET of the entire loop nest, the linear system

$$\begin{pmatrix} 0 \\ c_1 \\ c_2 \\ c_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 2 & 6 \\ 0 & 0 & 0 & 6 \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix}$$

is solved to obtain the coefficients of polynomial $p(\Phi, N)$

$$\begin{aligned} p_0 &= 0 \\ p_1 &= c_1 - \tfrac{1}{6} c_2 \\ p_2 &= 0 \\ p_3 &= \tfrac{1}{6} c_2 \end{aligned}$$

which are used to establish the WCET of the entire loop nest

$$\begin{aligned} WCET &= \max(c_0, c_0 + p(\Phi, N)) \\ &= \max(c_0, c_0 + (c_1 - \tfrac{1}{6} c_2) N + \tfrac{1}{6} c_2\, N^3) \quad . \end{aligned}$$

In contrast, conventional (parameterized) WCET analysis computes the maximum size of the inner loop iteration space by using the bounds $1 \leq I \leq N$ resulting in the iteration space bound $\lfloor \frac{N^2-1}{2} \rfloor$. Since the number of iterations of the outer loop is bounded by N, the parameterized WCET is

$$WCET = c_0 + c_1\, N + c_2\, N \lfloor \tfrac{N^2-1}{2} \rfloor$$

which is valid for $N > 1$.

Table 1 shows the WCET numbers of simple WCET compared to the proposed WCET estimation framework and a WCET Ω obtained by an exact count. The example uses the bounds $c_0 = c_1 = c_2 = 1$. The exact count is obtained by execution of the loop nest

```
Ω := c0
for I = 1 to N do
    Ω += c1
    for J = I to I*I-2 step 2 do
        Ω += c2
    od
od
```

N	Simple	New	Ω
1	2	2	2
5	66	26	26
10	501	176	176
50	62501	20876	20876
100	500001	166751	166751

Table 1. Comparison of WCET Estimation

Because the stride 2 evenly divides $i + i^2$, the proposed WCET estimation framework returns the exact count.

In some cases, a bound can be derived with Sakellariou's summation operator $\circledS$. However, the operator cannot be used for this example because it requires a summation over an expression guarded by a non-linear condition on the index variable i

$$
\begin{aligned}
&WCET \\
&= c_0 + \sum_{i=0}^{N-1}\left(c_1 + \sum_{j=0}^{\lfloor\frac{i+i^2}{2}\rfloor-1} c_2\right) \\
&= c_0 + \sum_{i=0}^{N-1}\left(c_1 + c_2\begin{cases}\lfloor\frac{i+i^2}{2}\rfloor & \text{if } 1 \leq \lfloor\frac{i+i^2}{2}\rfloor \\ 0\end{cases}\right) \quad .
\end{aligned}
$$

This cannot be further simplified [5] without sacrificing the accuracy of the WCET bound.

4. Conclusions

In this paper, we have proposed a framework for parametric WCET estimation and iteration counting for multiple rectangular and non-rectangular loop nests including those with zero-trip loops and non-unit strides. Symbolic bounds are produced via an algorithm based on our previous work on CR algebra [7] that only requires symbolic polynomial manipulation capabilities that amount to slight extensions of constant folding. Additionally, an alternative form of the computations that depends only on symbolic manipulation of pre-computed matrices was presented. The only restrictions on the multiple loop nest imposed by the framework are certain monotonicity properties of the growth of the iteration spaces of the inner loops.

The proposed framework is also applicable to generating bounds on power requirements for simple power models, counting instructions, and counting memory accesses.

Extensions are currently under consideration. These include the relaxation of the monotonic properties of the iteration spaces of the inner loops, characterizing the tightness of the bounds on WCET and other counts, and bounds for more complicated loop structures.

References

[1] O. Bachmann. *Chains of Recurrences*. PhD thesis, Kent State University of Arts and Sciences, 1996.

[2] T. Fahringer and B. Scholz. Symbolic evaluation for parallelizing compilers. In 11^{th} *ACM International Conference on Supercomputing*, pages 261–268, New York, 1997. ACM Press.

[3] M. R. Haghighat. *Symbolic Analysis for Parallelizing Compilers*. Kluwer Academic Publishers, 1995.

[4] C. Healy, M. Sjödin, V. Rustagi, D. Whalley, and R. van Engelen. Supporting timing analysis by automatic bounding of loop iterations. *Real-Time Systems*, pages 121–148, May 2000.

[5] R. Sakellariou. Symbolc evaluation of sums for parallelising compilers. In *IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics*. Wissenshaft & Technik Verlag, 1997.

[6] R. A. van Engelen. Efficient symbolic analysis for optimizing compilers. In *Proc. of the ETAPS Conference on Compiler Construction 2001, LNCS 2027*, pages 118–132, 2001.

[7] R. A. van Engelen and K. Gallivan. An efficient algorithm for pointer-to-array access conversion for compiling and optimizing dsp applications. In *2001 International Workshop on Innovative Architecture (IWIA 2001)*, pages 80–89, Maui, Hawaii, 2001.

[8] E. Vivancos, C. Healy, F. Mueller, and D. Whalley. Parametric timing analysis. In *LCTES 2001*, pages 88–93, 2001.

[9] E. Zima. Simplification and optimization transformations of chains of recurrences. In *Proc. of the International Symposium on Symbolic and Algebraic Computing*, Montreal, Canada, 1995. ACM.