Loop Acceleration under the presence of overflows [2]

Charles Babu M

Chennai Mathematical Institute

December 7, 2017

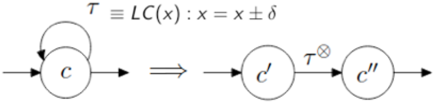
[2] Joint Work with M.K.Srivas and M. Praveen

Outline

- Loop Acceleration
- Overflow Examples
- Problems to be solved
- Program Termination
- Current Work

Loop Acceleration

int x = x0;
while (LC(x))
  $x = x \pm \delta$;



- Acceleration is typically restricted to programs over fragments of linear arithmetic for which the transitive closure is effectively computable
- Existing acceleration work restricted to operations on unbounded integers
- Challenges in handling integer overflows
  - Non-linear arithmetic and closed form
  - More complicated Termination check

Overflow: Program 1

```
int main(){
unsigned int x=0;
    while(x<=MAX_INT-1){
        x=x+2;
    }
}
```

Here, termination depends on the initial value of x. If x is even, it doesn't terminate as the wrap-around value of x repeats.

Overflow: Program 2

```
int main(){
unsigned int x=0;
    while(x<=MAX_INT-1){
        x=x+3;
    }
}
```

Here, termination is guaranteed. This is because, the MAX_INT is reachable from any initial value of x, as 3 and 2^{32} are relatively prime, which guarantees every wrap-around value generated is unique.

Our Program Template

Restricted Linear Programs (RLP) with conditionless loop body:

```
int main(){
unsigned int x=*;
  while(x<c){ //c is a constant
        x=x+delta; //'delta' is a constant
  }
  return 0;
}
```

Let $n = 2^w$, where w is the word length and $MAXINT = n - 1$.
Let delta(δ) is a constant, and without loss of generality consider $\delta > 0$. Let $d = |[c, MI]| > 0$

Problem Definition: Termination

- Termination: For a given x_0, it is an existential problem.

  $$\exists k.\ c \leq (x_0 + k \cdot \delta) \mod n \leq MI$$

  For the program to terminate, the following should be true

  $$\forall x \exists k.\ c \leq (x + k \cdot \delta) \mod n \leq MI$$

  - Model enumeration can be done as the domain is finite
  - Without model enumeration, can we give a necessary and sufficient condition for termination?

Accelerated Form Synthesis

- Accelerated Form Synthesis: If it indeed terminates, can we give a closed form?
  - For a given x_0 the exact 'k' after which it terminates is the minimum 'k' such that $(x_0 + k \cdot \delta) \mod n \in [c, MI]$
  - This minimization problem can also be expressed as an ILP problem

$$\text{minimize } k$$

$$c \leq x_0 - tn + k\delta \leq MI$$

$$k \geq 0, t \geq 0 \qquad k, t \in \mathbb{Z}$$

  - The synthesis problem requires one to generate a skolem expression as a function of x_0 for k s.t. that satisfies the above condition.

For now, we will address the termination problem.

RLP: Our Program Template

We will now look into termination aspect of the following program. Here, loop condition can be of the form $E \sim c$, where $\sim \in \{<, \leq, >, \geq, =, \neq\}$ and c a constant. Here `delta`(δ) > 0, $d = |[c, MI]| > 0$, and $n = 2^w$, where w is the word length.

```
int main(){
unsigned int x=*;
  while(x<c){ //c is a constant
        x=x+delta; //'delta' is a constant
  }
  return 0;
}
```

Termination: Our Main Result

For a program P $\in$ RLP, we have the following cases:

- $\delta \leq d$: No overflow and P terminates due to monotonicity of loop condition
- $\delta > d$:
  - $coprime(\delta, n)$: P terminates as we show $\exists k. x +_n k\delta \in [c, MI]$, where $+_n$ is modulo addition
  - $\neg coprime(\delta, n)$: Then, $\delta = 2^p \times \delta'$, where $e := 2^p, p \geq 0$ for some odd number δ'. Then:
    - If $e \leq d$: Program terminates and it will be proved
    - If $e > d$: Program terminates if and only if $(c \mod e) \leq x_0 \mod e \leq MI \mod e$

$\neg coprime(\delta, n)$

Let's analyze the case of $\delta > d$ and $\neg coprime(\delta, n)$. So, we have $\delta = 2^p \times \delta'$. Say, $e := 2^p$.
We will show that the following sets are equivalent

$$S_1 := \{x, x +_n \delta, x +_n 2 \cdot \delta, x +_n 3 \cdot \delta, ...\}$$

$$S_2 := \{x, x +_n e, x +_n 2 \cdot e, x +_n 3 \cdot e, ...\}$$

If the two sets are indeed equal, and if $x +_n k \cdot e \in [c, MI]$ for some k, then $x +_n k' \cdot \delta \in [c, MI]$ for some k', and so, the program terminates. The two sets S_1 and S_2 are finite.

Cyclic groups

Proposition

The additive group $(\mathbb{Z}_n, +)$ is a cyclic group of order n with (additive) generator 1.

Note that there is only one cyclic group of order n, up to isomorphism. So any statement about the additive groups $(\mathbb{Z}_n, +)$ is a statement about finite cyclic groups, and vice versa.

Proposition

In a finite cyclic group, two elements generate the same subgroup if and only if the elements have the same order.

Here $gcd(n, \delta) = gcd(n, e)$, so the subgroups of $(\mathbb{Z}_n, +)$ generated by the elements e and δ are the same

Equivalence between sets S_1 and S_2

Corollary

For some $0 \leq f < n$,
$k \cdot \delta \mod n \equiv f$ *holds for some* $k \iff k' \cdot e \mod n \equiv f$ *holds for some* k'

Corollary

For some $0 \leq f < n$,
$(x + k \cdot \delta) \mod n \equiv f$ *holds for some* $k \iff (x + k' \cdot e) \mod n \equiv f$ *holds for some* k'

It follows that the two sets S_1 and S_2 are equivalent.

Termination condition: $\neg coprime(\delta, n)$

Theorem

$e \leq d \implies$ *The program terminates.*

Proof.

Proof is simple. Here, all we need to show is that the set S_2 contains an element of $[c, MI]$.

- $x \geq c$: we're done
- $x < c$: As $e > 0$,

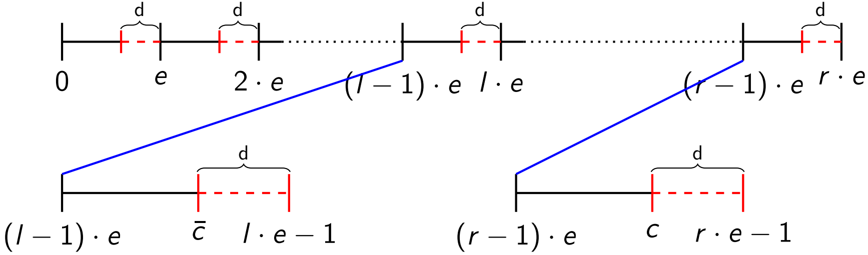
$$\exists k \ \text{ s.t. } \ (x + k \cdot e < c) \wedge (x + (k+1) \cdot e \geq c)$$

So, as $x + k \cdot e < c$, we have $x + k \cdot e + e < c + e$. Also, as $e \leq d$, we have $c \leq x + (k+1) \cdot e < MI + 1$. Hence, an element of $[c, MI]$ is in the set S_2.

□

Termination condition: $\neg coprime(\delta, n)$

$e > d$: Consider the set S_2. Let $r := \frac{n}{e} = 2^{w-p}$, and we have $MI = r.e - 1$. Here, $\bar{c} = c - (r - l) \cdot e$.



As $e|n$, if the initial configuration of x is in a red region, then the set S_2 contains a unique element of every other red region. If not, the set doesn't any element from any red region.

Theorem

The program terminates if and only if $(c \mod e) \leq x \mod e \leq MI \mod e$

Future Work

- Can we lift the problem to higher dimension with the termination condition being a convex polyhedron?
- Can we relax the variable assignment with δ not being a constant?
  - E.g. for assignment such as in the following program

$$y = *; while(x < c)\{x = x + y;\}$$

    As $n = 2^w$, it is enough to consider w subgroups of $(\mathbb{Z}_n, +)$ generated by $2^0, 2^1, ..., 2^{w-1}$, to see whether the program terminates or not.