ATVA 2023 · 25 October 2023

pymwp: A Static Analyzer Determining Polynomial Growth Bounds

Clément Aubert, Thomas Rubiano, **Neea Rusch** and Thomas Seiller

```
void main(int X1, int X2, int X3)
{
  while(X1 < 10){
     X1 = X2 + X3;
  }
  // X1'  X2'  X3'
}
```

$\forall n$, is $\mathrm{X}_n \rightsquigarrow \mathrm{X}'_n$ polynomially bounded in inputs?

```
void main(int X1, int X2, int X3)
{
  while(X1 < 10){
     X1 = X2 + X3;
  }
  // X1'  X2'  X3'
}
```

Yes, here is a bound:

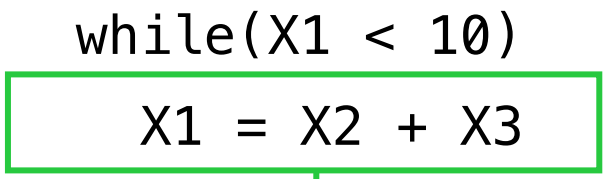
$$X1' \leq \max(X1, X2+X3)$$

$$\wedge\ X2' \leq X2$$

$$\wedge\ X3' \leq X3$$

mwp-flow analysis

Calculus for resource analysis of imperative programs.



$$\begin{pmatrix} m & 0 & 0 \\ p & m & 0 \\ m & 0 & m \end{pmatrix}$$

0 – no dependency

m – maximal (of linear)

w – weak polynomial

p – polynomial

Neil D. Jones and Lars Kristiansen. “A flow calculus of mwp-bounds for complexity analysis”. In: ACM Trans. Comput. Log. 10.4 (2009).

mwp-flow analysis

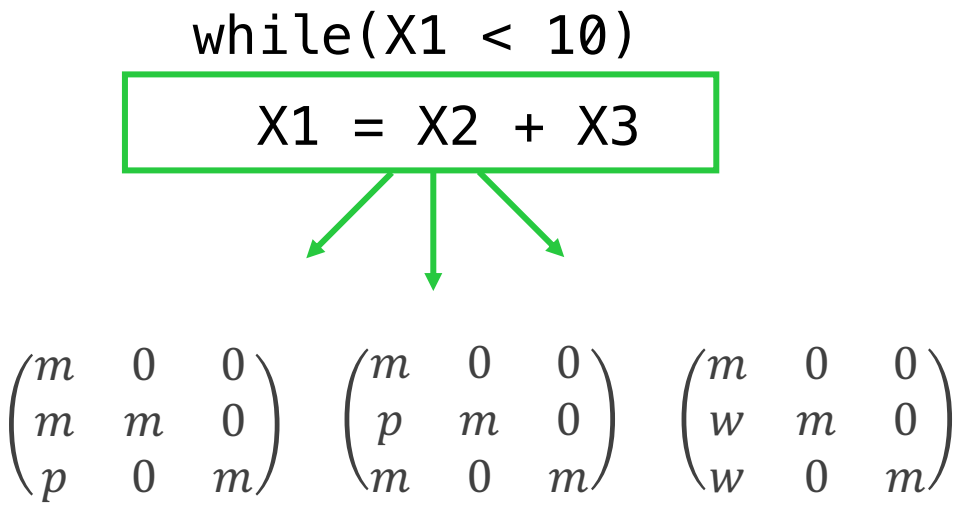
Calculus for resource analysis of imperative programs.

0 – no dependency

m – maximal (of linear)

w – weak polynomial

p – polynomial

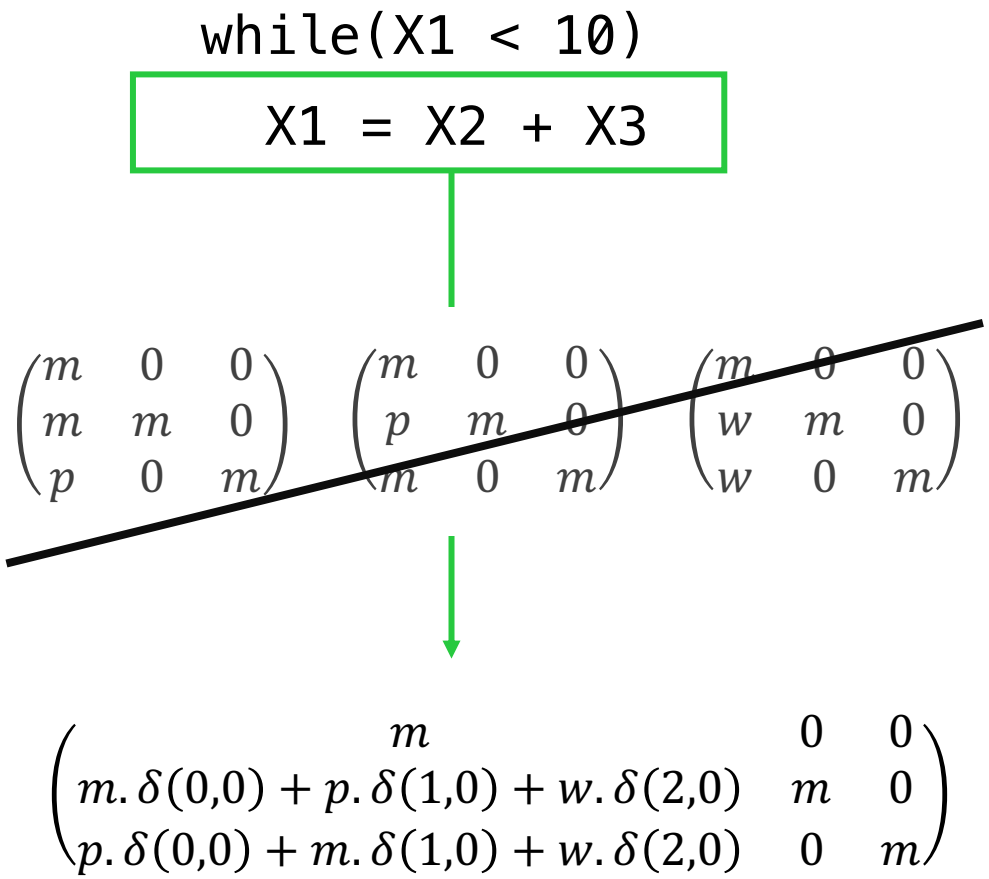


$$\begin{pmatrix} m & 0 & 0 \\ m & m & 0 \\ p & 0 & m \end{pmatrix} \quad \begin{pmatrix} m & 0 & 0 \\ p & m & 0 \\ m & 0 & m \end{pmatrix} \quad \begin{pmatrix} m & 0 & 0 \\ w & m & 0 \\ w & 0 & m \end{pmatrix}$$

Neil D. Jones and Lars Kristiansen. “A flow calculus of mwp-bounds for complexity analysis”. In: ACM Trans. Comput. Log. 10.4 (2009).

Automating mwp

Internalize non-determinism

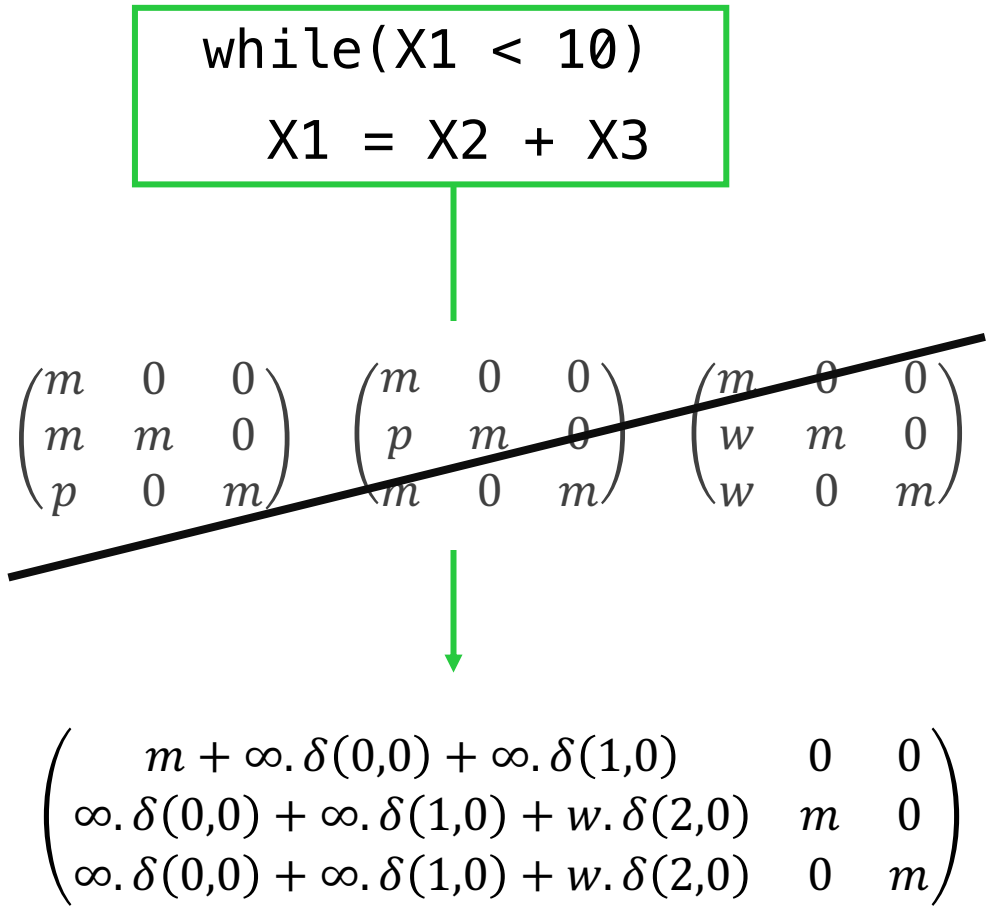


Automating mwp

Handle derivation failure

$$0, m, w, p, \infty$$

∞ - non-polynomial / failure



```
void main(int X1, int X2, int X3)
{
  while(X1 < 10){
     X1 = X2 + X3;
  }
  // X1'  X2'  X3'
}
```

We were here

$$\begin{pmatrix} m + \infty.\delta(0,0) + \infty.\delta(1,0) & 0 & 0 \\ \infty.\delta(0,0) + \infty.\delta(1,0) + w.\delta(2,0) & m & 0 \\ \infty.\delta(0,0) + \infty.\delta(1,0) + w.\delta(2,0) & 0 & m \end{pmatrix}$$

But we want

$X1' \leq \max(X1, X2+X3)$
$\wedge\ X2' \leq X2\ \wedge\ X3' \leq X3$

```
void main(int X1, int X2)
{
  X1 = X2 + X2;
  while(X1 < 10){
     X1 = X1 * X1;
  }
}
```

When derivation fails

Problematic flows:

$X1 \rightarrow X1$

$X2 \rightarrow X1$

pymwp is an automatic static analyzer for subset of C code,
to determine if variables' value growth is polynomially bounded.

run in terminal

```
pip install pymwp

pymwp file.c
```

run in browser

`statycc.github.io/pymwp/demo`