

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'
}
```

Does there exist a polynomially  
bounded data-flow relation  
between variables'  
**initial** and **final** values?

$\forall n$ , is  $X_n \rightsquigarrow X'_n$   
polynomially bounded?



```
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<sup>1</sup>

Calculus for resource analysis of imperative programs.

0 – no dependency


$m$  – maximal (of linear)

$w$  – weak polynomial

$p$  – polynomial

while ( $X1 < 10$ )

$X1 = X2 + X3$


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

---

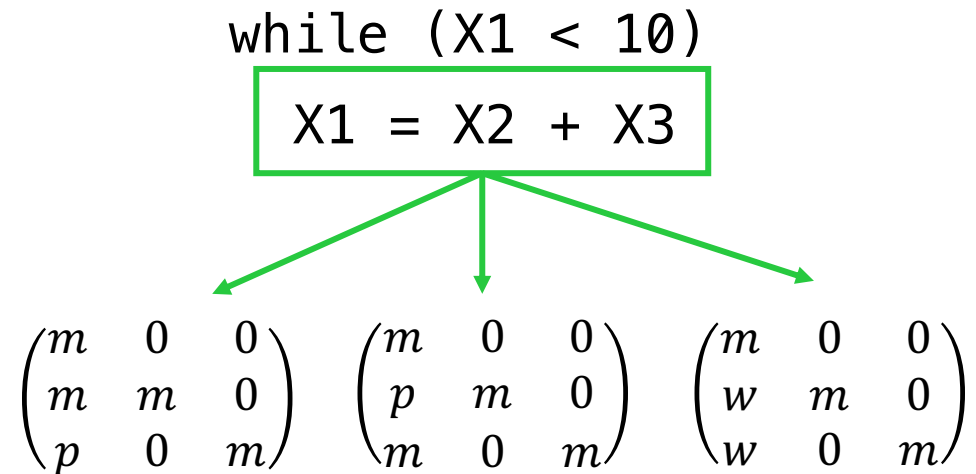
<sup>1</sup>Neil D. Jones and Lars Kristiansen. “A flow calculus of mwp-bounds for complexity analysis”. In: ACM Trans. Comput. Log. 10.4 (Aug. 2009), 28:1–28:41. doi: [10.1145/1555746.1555752](https://doi.org/10.1145/1555746.1555752).

# mwp-flow analysis<sup>1</sup>

Problems:

Non-determinism

Derivation failure

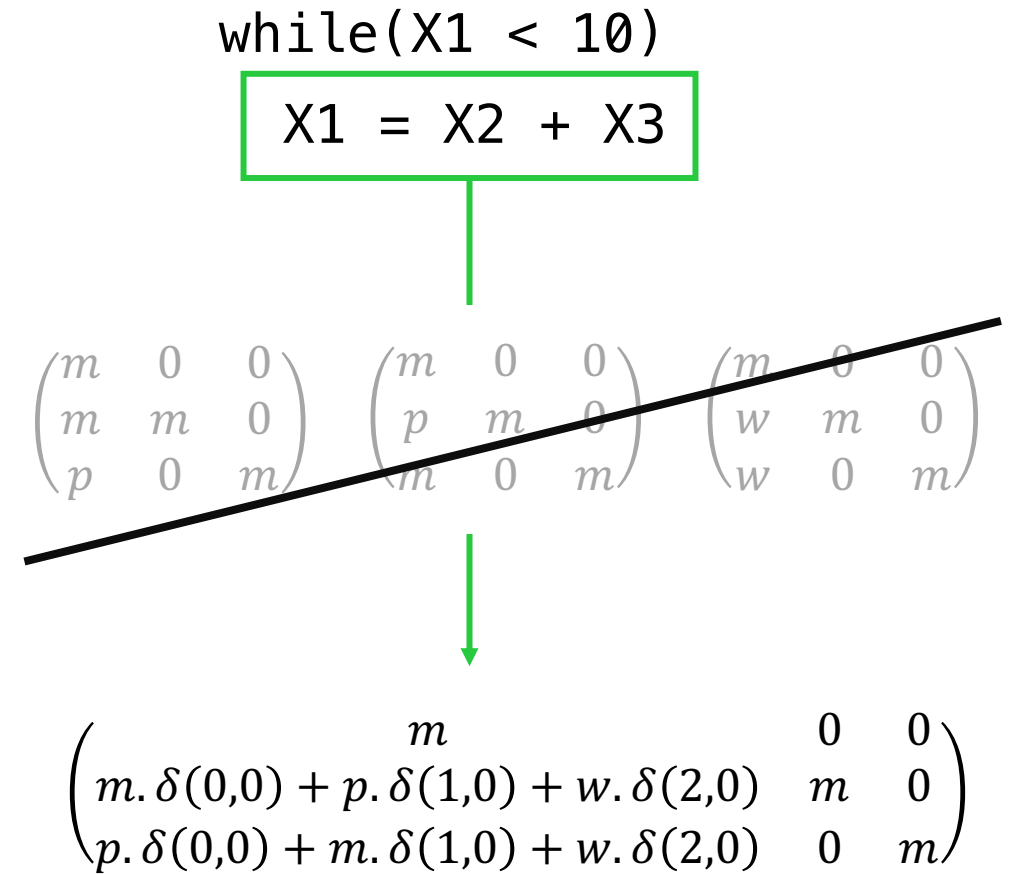


---

<sup>1</sup>Neil D. Jones and Lars Kristiansen. “A flow calculus of mwp-bounds for complexity analysis”. In: ACM Trans. Comput. Log. 10.4 (Aug. 2009), 28:1–28:41. doi: [10.1145/1555746.1555752](https://doi.org/10.1145/1555746.1555752).

# Automating mwp

- ✓ Internalize non-determinism

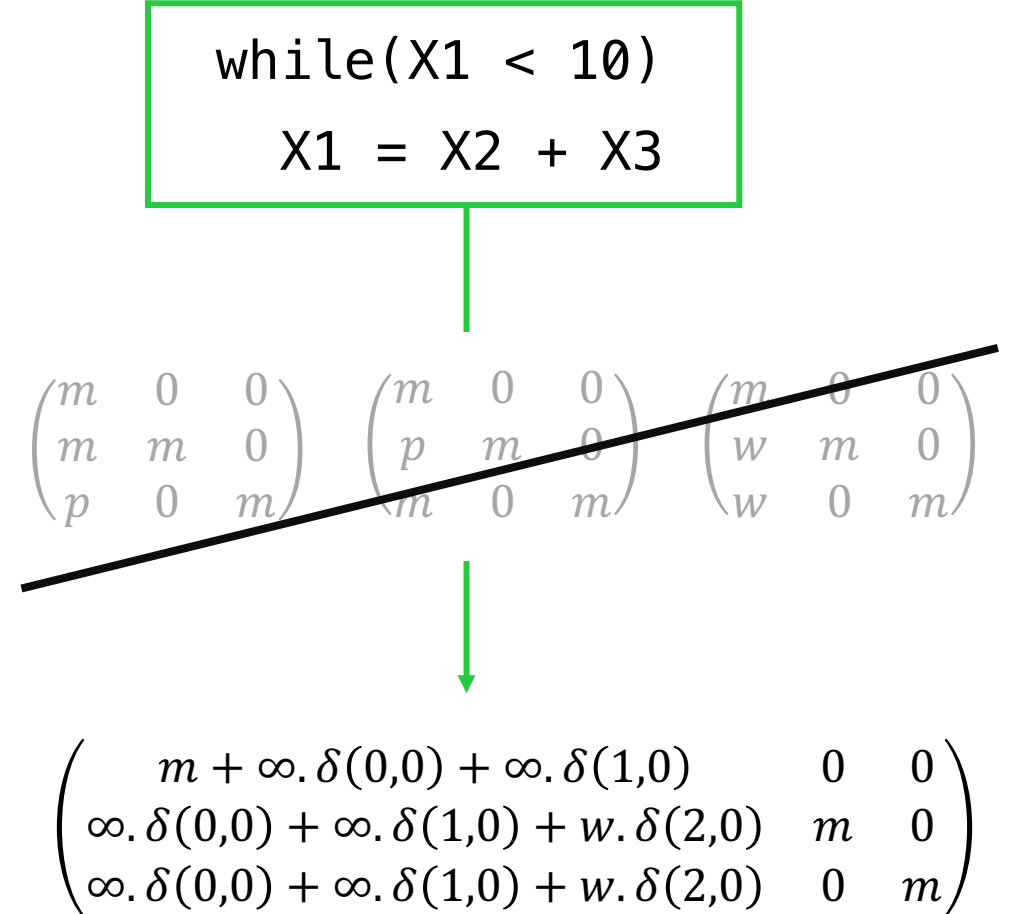


# Automating mwp

- ✓ Internalize non-determinism
- ✓ Handle derivation failure

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

$\infty$  - non-polynomial / failure





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

    // X1'   X2'   X3'

}
```

We were here

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

But we want

$$\begin{aligned} X1' &\leq \max(X1, X2+X3) \\ \wedge X2' &\leq X2 \quad \wedge X3' \leq X3 \end{aligned}$$



# Obtaining bounds

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

Find the derivation choices that lead to  $\infty$ , then simplify

$$[\{\cancel{0}, \cancel{1}, 2\}] \rightarrow [\{2\}]$$

Exclude failing choices

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

Final bound



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


When derivation fails

Problematic flows:

$X1 \rightarrow X1 \parallel X2 \rightarrow X1$

**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](https://statycc.github.io/pymwp/demo)



statycc/pymwp