Co-author Declaration

A summary of work performed by Thomas Rubiano on "pymwp: A Static Analyzer Determining Polynomial Growth Bounds."

Thomas Rubiano is the original software architect of the pymwp analyzer and responsible for much of the groundwork. His contributions included crafting the overall analysis workflow, and designing the core data structures. For example, the Relation data structure (discussed in Sect. 4.1), and Polynomial structures, that track coefficients in the matrices, where originally implemented by Thomas. The starting point of the pymwp analyzer is based on earlier work on Loop peeling with Thomas Seiller and Jean-Yves Moyen (`https://github.com/statycc/LQICM_On_C_Toy_Parser`). The software development contributions to pymwp are detailed at:
`https://github.com/statycc/pymwp/graphs/contributors`.

Signé par :
Thomas Rubiano
1950A0F86415413...

5/28/2025 | 06:48:38 EDT

Thomas Rubiano

Date signed

Co-author Declaration

A summary of work performed by Thomas Rubiano on "Distributing and Parallelizing Non-canonical Loops."

Thomas's Rubiano's doctoral work (Rubiano, 2017) on compositional data flow static analysis in compilers is a grounding motivation behind this work. Thomas Rubiano and Thomas Seiller collaborated on the initial concept of splitting loops by dependencies, and the mathematical analysis behind it, particularly during our seminar in Dagstuhl in 2021 (https://www.dagstuhl.de/en/seminars/seminar-calendar/seminar-details/21453). Thomas Rubiano created a prototype analyzer implementing the theory, a test environment and benchmarks to test the theory. Although the analyzer and those benchmarks did not make it to the final paper (https://github.com/statycc/loop-fission/graphs/contributors), their development facilitated the development of the end result. For example, the loop fission algorithm (Sect. 3) refined original ideas behind the prototype.

Signé par :

Thomas Rubiano

1950A0F86415413...

Thomas Rubiano

5/28/2025 | 06:48:38 EDT

Date signed

Co-author Declaration

A summary of work performed by Thomas Rubiano on "mwp-Analysis Improvement and Implementation: Realizing Implicit Computational Complexity."

All authors actively discussed ideas on how to improve and implement the flow calculus during the research meetings. Thomas Rubiano implemented the delta graph technique (Sect. 4.4) and contributed to the implementation (Sect. 5 and Appendix C). Precise contributions to the source code can be traced using `https://github.com/statycc/pymwp/graphs/contributors`.

Signé par :

Thomas Rubiano

1950A0F86415413...

5/28/2025 | 06:48:38 EDT

Thomas Rubiano　　　Date signed