

Aligator: Experimental Results on Textbook Examples

Laura Kovács

EPFL, Switzerland
laura.kovacs@epfl.ch

Table 1 reports on experimental results of `Aligator` on some textbook examples.

The first column of the table contains the name of the example, the second and third columns specify the applied combinatorial methods and the number of generated polynomial invariants for the corresponding example, whereas the fourth column shows the timing (in seconds) needed by the implementation on a machine with 2.0GHz CPU and 2GB of memory. The fifth column shows whether our method was complete.

Example	Comb. Methods	Nr.Poly.	(sec)	Compl.
P-solvable loops with assignments only				
Division [1]	Gosper	1	0.031	yes
Integer square root [3]	Gosper	2	0.063	yes
Integer square root [4]	Gosper	1	0.046	yes
Integer cubic root [4]	Gosper	2	0.094	yes
Fibonacci [5]	C-finite, Alg.Dependencies	1	0.219	yes
Sum of powers n^5 [7]	Gosper	1	0.125	yes
P-solvable loops with conditional branches and assignments				
Wensley's Algorithm [10]	Gosper, C-finite, Alg.Dependencies	3	0.25	yes
LCM-GCD computation [1]	Gosper	1	0.437	yes
Extended GCD [4]	Gosper	5	3.094	yes
Fermat's factorization [4]	Gosper	1	0.109	yes
Square root [11]	Gosper, C-finite, Alg.Dependencies	1	0.406	yes
Binary Product [4]	Gosper, C-finite, Alg.Dependencies	1	0.219	yes
Binary Product [8]	Gosper, C-finite, Alg.Dependencies	1	0.297	yes
Binary Division (2nd Loop) [2]	C-finite, Alg. Dependencies	1	0.219	yes
Hardware Integer Division [6]	Gosper, C-finite, Alg.Dependencies	3	0.25	yes
Hardware Integer Division [9]	Gosper, C-finite, Alg.Dependencies	3	0.25	yes
Factoring Large Numbers [4]	C-finite, Gosper	1	0.906	yes

Table 1. Experimental Results

References

1. E. W. Dijkstra. *A Discipline of Programming*. Prentice-Hall, 1976.
2. A. Kaldewaij. *Programming. The Derivation of Algorithms*. Prentice-Hall, 1990.
3. M. Kirchner. Program Verification with the Mathematical Software System Theorema. Technical Report 99-16, RISC-Linz, Austria, 1999. Diplomaarbeit.
4. D. E. Knuth. *The Art of Computer Programming*, volume 2. Seminumerical Algorithms. Addison-Wesley, 3rd edition, 1998.
5. L. Kovács. *Automated Invariant Generation by Algebraic Techniques for Imperative Program Verification in Theorema*. PhD thesis, RISC, Johannes Kepler University Linz, 2007.
6. Z. Manna. *Mathematical Theory of Computation*. McGraw-Hill Inc., 1974.
7. M. Petter. Berechnung von polynomiellen Invarianten. Master's thesis, Technical University Munich, Germany, 2004.
8. E. Rodriguez-Carbonell and D. Kapur. Generating All Polynomial Invariants in Simple Loops. *J. of Symbolic Computation*, 42(4):443–476, 2007.
9. S. Sankaranaryanan, H. B. Sipma, and Z. Manna. Non-Linear Loop Invariant Generation using Gröbner Bases. In *Proc. of POPL 2004*, 2004.
10. B. Wegbreit. The Synthesis of Loop Predicates. *Communication of the ACM*, 2(17):102–112, 1974.
11. K. Zuse. *The Computer - My Life*. Springer, 1993.