# 1 Logical Interpretation

Abstract Interpretation over Logical Formulas. We will consider (conjunctive) logical formulas over the theories of linear arithmetic, uninterpreted functions, and their combination.

Proposal: The bold stuff will be covered in relatively more detail, while the rest of the stuff will be described relatively briefly. The four main references that will be followed are: [5, 7, 8, 2].

1. Decision Procedures
   - Linear Arithmetic
     - Equalities (Gaussian Elimination)
     - **Difference Constraints** (Transitive Closure)
     - Inequalities (Simplex)
   - **Uninterpreted Functions** (Congruence Closure)
   - **Combination of Linear Arithmetic and Uninterpreted Functions** (Nelson-Oppen [16])
2. Hardness Results for Assertion Checking
   - Linear Arithmetic [14]
   - Uninterpreted Functions [13, 5]
   - **Combination of Linear Arithmetic and Uninterpreted Functions** [7, 9]
3. Intra-procedural Invariant Discovery
   - Linear Arithmetic
     - Equalities [11, 14, 3]
     - **Difference Constraints** [12]
     - Inequalities [1]
   - **Uninterpreted Functions** [5, 10, 4]
   - **Combination of Linear Arithmetic and Uninterpreted Functions** [8]
   - **Arrays**: Universally Quantified Formulas over Combination of Linear Arithmetic and Uninterpreted Functions [2]
4. Other aspects
   - Inter-procedural Invariant Discovery
     - Linear Arithmetic: Linear Constants [17], Equalities [15, 6]
   - Disjunctive/Power-set domains

# References


[1] P. Cousot and N. Halbwachs. Automatic discovery of linear restraints among variables of a program. In *POPL*, pages 84–97, 1978.

[2] S. Gulwani, B. McCloskey, and A. Tiwari. Lifting abstract interpreters to quantified logical domains. Technical Report MSR-TR-2007-87, Microsoft Research, July 2007.

[3] S. Gulwani and G. C. Necula. Discovering affine equalities using random interpretation. In *The 30th Annual ACM Symposium on Principles of Programming Languages*, pages 74–84. ACM, Jan. 2003.

[4] S. Gulwani and G. C. Necula. Global value numbering using random interpretation. In *31st ACM Symposium on Principles of Programming Languages*, pages 342–352. ACM, Jan. 2004.

[5] S. Gulwani and G. C. Necula. A polynomial-time algorithm for global value numbering. In *11th Static Analysis Symposium*, volume 3148 of *LNCS*, pages 212–227. Springer-Verlag, Aug. 2004.

[6] S. Gulwani and G. C. Necula. Precise interprocedural analysis using random interpretation. In *The 32nd Annual ACM Symposium on Principles of Programming Languages*. ACM, Jan. 2005.

[7] S. Gulwani and A. Tiwari. Assertion checking over combined abstraction of linear arithmetic & uninterpreted functions. In *ESOP*, volume 3924 of *LNCS*, Mar. 2006.

[8] S. Gulwani and A. Tiwari. Combining abstract interpreters. In *PLDI*, pages 376–386, June 2006.

[9] S. Gulwani and A. Tiwari. Assertion checking unified. In *The 8th International Conference on Verification, Model Checking and Abstract Interpretation*. Springer, Jan. 2007.

[10] S. Gulwani, A. Tiwari, and G. C. Necula. Join algorithms for the theory of uninterpreted symbols. In *Conf. Found. of Soft. Tech. and Theor. Comp. Sci., FST&TCS '2004*, volume 3328 of *LNCS*, pages 311–323, 2004.

[11] M. Karr. Affine relationships among variables of a program. In *Acta Informatica*, pages 133–151. Springer, 1976.

[12] A. Miné. A new numerical abstract domain based on difference-bound matrices. In *PADO '01: Proceedings of the Second Symposium on Programs as Data Objects*, pages 155–172. Springer-Verlag, 2001.

[13] M. Müller-Olm, O. Rüthing, and H. Seidl. Checking herbrand equalities and beyond. In *VMCAI*, volume 3385 of *LNCS*, pages 79–96. Springer, Jan. 2005.

[14] M. Müller-Olm and H. Seidl. A note on Karr's algorithm. In *31st International Colloquium on Automata, Languages and Programming*, pages 1016–1028, 2004.

[15] M. Müller-Olm and H. Seidl. Precise interprocedural analysis through linear algebra. In *31st ACM Symposium on POPL*, pages 330–341, Jan. 2004.

[16] G. Nelson and D. Oppen. Simplification by cooperating decision procedures. *ACM Transactions on Programming Languages and Systems*, 1(2):245–257, Oct. 1979.

[17] M. Sagiv, T. Reps, and S. Horwitz. Precise interprocedural dataflow analysis with applications to constant propagation. *Theor. Comput. Sci.*, 167(1-2):131–170, 1996.