**loop problem=> invariant generation problem => constraint solving problem**

**Transition Systems and Invariants**

**A transition system P : (V, L, l0, Θ, T)**

a set of variables V

a set of locations L

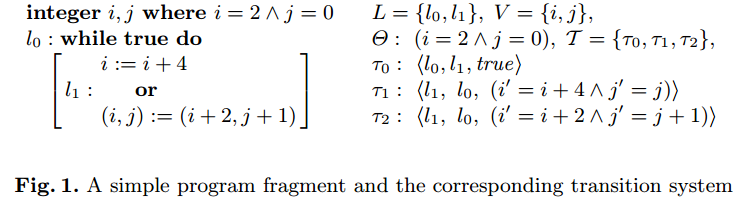
an initial location l0

an initial assertion Θ over the variables V

a set of transitions T

Each transition τ ∈ T is a tuple (l, l0, ρτ), where l, l0 ∈ L are the pre and post locations, and ρτ is the transition relation,

an assertion over V ∪V0, where V represents current-state variables and its primed version V0 represents the next-state variables.



**Linear and Non-Linear Constraints**

A linear constraint over V is an inequality of the form a1x1 +· · ·+anxn +b ≤ 0,

where a1, · · · , an, b denote known real-valued coeﬃcients

A non-linear constraint is an inequality of the form P ≤ 0,

where P is a polynomial on x1, . . ., xn.

**Generating Invariants**

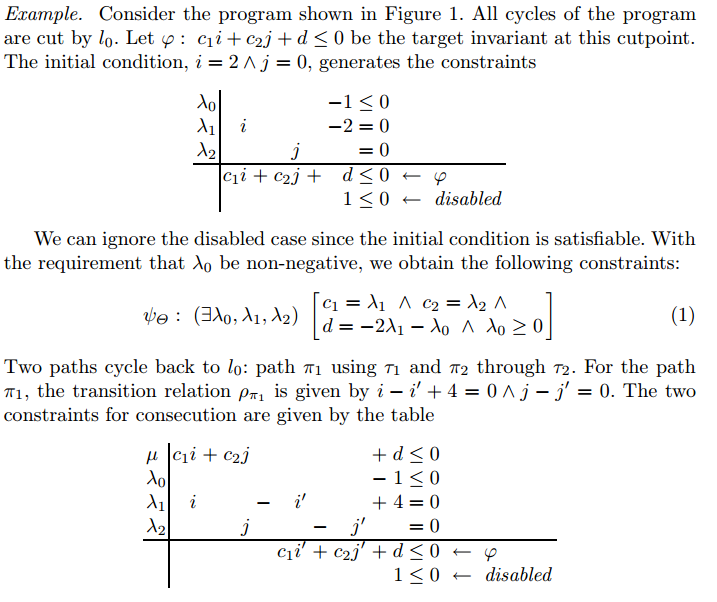
**Farkas’ Lemma**

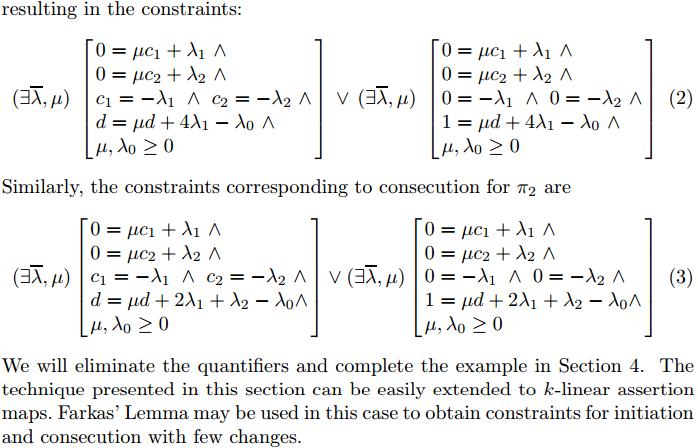
(S CHRIJVER, A. Theory of Linear and Integer Programming. Wiley, 1986.)

**sound and complete**

any linear invariant of a linear program which is provable using an inductive linear assertion can be be found

**The main drawback of the method is that it produces non-linear constraints**





**Fixpoint**

COUSOT, P., AND COUSOT, R. **Abstract Interpretation**: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In ACM Principles of Programming Languages (1977), pp. 238–252.

**Constraint-based technique to generate linear invariants**

COLON, M., S ANKARANARAYANAN, S., AND SIPMA , H. **Linear invariant generation using non-linear constraint solving**. In Computer Aided Verification (July 2003), F. Somenzi and W. H. Jr, Eds., vol. 2725 of LNCS, Springer Verlag, pp. 420–433

**Gr¨obner bases**

C OX, D., LITTLE, J., AND O’SHEA, D. **Ideals, Varieties and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra**. Springer, 1991.

Gr¨obner bases can be used to **reduce the invariant generation problem to a non-linear constraint solving problem** that is shown to be in the parametric linear form (BALLARIN, C., AND KAUERS, M. Solving parametric linear systems: an experiment with constraint algebraic programming. In Eighth Rhine Workshop on Computer Algebra (2002), pp. 101–114.)

**hybrid systems**

Sankaranarayanan, S., Sipma, H.B., Manna, Z.: **Constructing invariants for hybrid systems**. Form. Methods Syst. Des. 32(1) (2008) 25{55

**Solving Constraints**

**Linear Constraints**

**simplex method** (A. Schrijver. Theory of Linear and Integer Programming. Wiley, 1986)

**double description** method (K. Fukuda and A. Prodon. Double description method revisited. In Combinatorics and Computer Science, volume 1120 of LNCS, pages 91–111. Springer-Verlag, 1996)

Alternatively, projection and the computation of generators can be achieved through a quantifier elimination method called **Fourier’s elimination**, which eliminates variables from the system of constraints incrementally (A. Bockmayr and V. Weispfenning. Solving numerical constraints. In A. Robinson and A. Voronkov, editors, Handbook of Automated Reasoning, volume I, chapter 12, pages 751–842. Elsevier Science, 2001.). Due to its simplicity, Fourier’s elimination has been used widely to solve linear constraints, even though its complexity is exponential.

**Non-linear Constraints** Non-linear constraints can be solved by direct quantiﬁer elimination or indirect methods using techniques such as factorization and polynomial root solving.

Quantifier elimination (A. Tarski. A decision method for elementary algebra and geometry. Univ. of California Press, Berkeley, 5, 1951)

**Cylindrical Algebraic Decomposition**

QEPCAD (G. E. Collins and H. Hong. Partial cylindrical algebraic decomposition for quantifier elimination. Journal of Symbolic Computation, 12(3):299–328, sep 1991) => time complexity

factorization and Gr¨obner Bases

REDLOG (A. Dolzmann and T. Sturm. REDLOG: Computer algebra meets computer logic. ACM SIGSAM Bulletin, 31(2):2–9, June 1997)

RISC-CLP(R) (H. Hong. RISC-CLP(Real): Constraint logic programming over real numbers. In CLP: Selected Research. MIT Press, 1993.)

CLP - J. Jaffar and J.-L. Lassez. Constraint logic programming. In Principles of Programming Languages(popl), pages 111–119, January 1987.

