

Enumerating Integer Projections of Parametric Polytopes: A Direct Approach

Sven Verdoolaege, Maurice Bruynooghe

Many compiler techniques depend on the ability to count the number of integer points that satisfy a given set of linear inequalities. Typically, a subset of the variables are identified as the *parameters* \mathbf{p} and the number of possible values for the other, “counted” variables \mathbf{y} needs to be counted as a function of those parameters. If each variable is either a parameter or counted, then the problem is equivalent to the enumeration of parametric polytopes, which can be computed efficiently using a technique based on Barvinok’s decomposition of unimodular cones [1, 3, 5].

In general, the linear inequalities may also involve additional, existentially quantified variables ϵ and then the object is to count the number of possible integer values for \mathbf{y} in function of the parameters \mathbf{p} such that there exist integers ϵ that satisfy the inequalities. This problem is equivalent to enumerating an integer projection of a parametric polytope. Pugh [4] addresses the closely related problem of counting the number of solutions to Presburger formulas, but his technique seems underspecified and has apparently never been implemented. One of the substeps is also clearly exponential in the input size. More recently, a technique was outlined for integer projections of non-parametric polytopes [2], but it appears not to have been implemented yet and it is not immediately clear how easily it can be extended to the parametric case.

Consider the polyhedron P defined by the given set of linear inequalities in $(d+d'+n)$ -dimensional space, where d , d' and n are the number of counted variables, existential variables and parameters. Our technique manipulates this polyhedron directly through a set of simplification rules, which either reduce the number of existential variables, shrink the polyhedron or split the problem into several smaller subproblems.

Pick an existential variable ϵ_i and take a pair of inequalities such that the coefficients of ϵ_i have opposite signs. The two inequalities determine a range for ϵ_i as a function of the other variables. If no such pair exists or if this range admits at least one integer so-

lution over the whole of P , then ϵ_i can be eliminated. If a pair exists such that at least one of the inequalities is independent of the other existential variables and such that the range admits at most one solution, then ϵ_i can be treated as a counted variable.

The polyhedron can be shrunk by adding cutting planes such that the bounding box of P has integer vertices. In principle, planes can be added until the integer hull is obtained, basically solving the problem, but this would be prohibitively expensive. Rays and lines can be removed from P by taking a slice from P and periodically extending the resulting enumeration.

In general, P will need to be split such that each part is closer to removal of an existential variable. If the splitting constraint is independent of the existential variables, the enumerations of both parts can simply be summed. Otherwise, we need to take the disjunction of both enumerations E_1 and E_2 . If $d = 0$, this can be computed as $E_1 + E_2 - E_1 E_2$. If $d \neq 0$, we first treat all counted variables as parameters and then sum the resulting enumeration over the counted variables.

References

- [1] A. Barvinok and J. Pommersheim. An algorithmic theory of lattice points in polyhedra. *New Perspectives in Algebraic Combinatorics*, 38:91–147, 1999.
- [2] A. Barvinok and K. Woods. Short rational generating functions for lattice point problems, Nov. 2002.
- [3] J. A. De Loera, R. Hemmecke, J. Tauzer, and R. Yoshida. Effective lattice point counting in rational convex polytopes, Mar. 2003.
- [4] W. Pugh. Counting solutions to Presburger formulas: How and why. In *Proc. PLDI'94*, pages 121–134, 1994.
- [5] S. Verdoolaege, R. Seghir, K. Beyls, V. Loechner, and M. Bruynooghe. Analytical computation of Ehrhart polynomials: enabling more compiler analyses and optimizations. In *Proc. CASES'04*.