Lazard's CAD Exploiting Equality Constraints

Akshar Nair
University of Bath, UK
asn42@bath.ac.uk

James Davenport
University of Bath, UK
masjhd@bath.ac.uk

Gregory Sankaran
University of Bath, UK
masgks@bath.ac.uk

Scott McCallum
Department of Computing, MacQuarie University, Australia
scott.mccallum@mq.edu.au

Abstract

McCallum improved the original Collins CAD projection operator (assuming well-orientation) and reduced the projection set even further for quantifier elimination problems which have equality constraints [6]. Lazard provided a projection operator (and corresponding lifting process) that reduces the projection set as compared to McCallum's and is unconditional like Collins' original algorithm [2]. Our research extends Lazard's work by providing a modification that reduces the projection set even further when there is an equality constraint in the quantifier elimination problem. We also report a slight error in [7].

1 Introduction

A Cylindrical Algebraic Decomposition (CAD) is a decomposition of a semi-algebraic set $S \subseteq \mathbb{R}^n$ into semi-algebraic sets homeomorphic to $\mathbb{R}^m$ (also known as cells) such that the projection of any two cells onto the first k coordinates is either the same or disjoint. Many algorithms for CAD consist of the following three phases.

Projection phase: Reduce the number of variables of the polynomial constraints (using a function known as the projection operator) until it has reached polynomials in one variable ($\mathbb{R}[x_1]$). The variable ordering is important (generally $x_1 > x_2 > \ldots > x_n$) i.e. the first variable eliminated is x_n.

Base phase: Decompose $\mathbb{R}^1$ according to specifications of the required CAD. Each cell is given a sample point, which is used for tracking cells in the lifting phase.

Lifting phase: Decompose higher dimensional spaces using the sample points and the projection polynomials in that dimension until S is decomposed.

Definition 1 *Let A be a set of polynomials in $\mathbb{R}[x_1, \ldots, x_n]$ and $P\colon \mathbb{R}[x_1, \ldots, x_n] \times \mathbb{R}^n \to \Sigma$ a function to some set Σ. If $C \subset \mathbb{R}^n$ is a cell and $P(f, \alpha)$ is independent of $\alpha \in C$ for every $f \in A$, then A is called P-invariant over that cell. If this is true for all cells of a decomposition, we say the decomposition is P-invariant for A.*

Collins [1] provided the first CAD algorithm to perform Quantifier Elimination (QE) over real fields, giving sign-invariant CADs. McCallum [5] provided the following projection operator, which gives order-invariant CADs of $\mathbb{R}^n$.

Definition 2 *Let A be a set of polynomial constraints. McCallum's projection operator $P(A)$ is the union of: the set of all non-zero coefficients of polynomials in A; the set of all discriminants of polynomials in A; the set of resultants of all distinct pairs of polynomials in A.*

$P(A)$ is smaller than Collins' projection operator, thus reducing the complexity. Unfortunately, the algorithms for computing CADs of $\mathbb{R}^n$ tend to be very expensive. If the input consists of m polynomials of at most degree d in n variables, the complexity of McCallum's algorithm is of the order $(2dm)^{2^{O(n)}}$ [3]. However McCallum later gave a reduced projection operator [6] when there are equational constraints in A, which will be seen in the next section.

Lazard [4] provided a different approach: rather than using the order of polynomials, he proposed using the lex-least valuation, derived from the expansion in local coordinates. This allowed a slightly smaller projection operator but required a significantly different lifting process. Our idea is to use lex-least invariance, following [4] to reduce the projection in the same way as in [6]. The next step would be to extend to[7], where we report a slight error. Further details are in [11].

2 Basic Terminology and McCallum's work

In this section, we explore McCallum's modifications to his original algorithm, exploiting equational constraints.

Definition 3 *[3] A Quantifier Free Tarski Formula (QFF) is made up of atoms connected by the standard boolean operators $\wedge, \vee$ and $\neg$. The atoms are statements about signs of polynomials with integer coefficients $f \sim 0$ where $\sim \in \{=, <, >\}$ (and by combination also $\{\geq, \leq, \neq\}$).*

Definition 4 *[3] An Equational Constraint (EC) is a polynomial equation logically implied by a QFF. If it is an atom of the formula, it is said to be* explicit*; if not, then it is* implicit*. If the constraint is visibly an equality one from the formula, i.e. the formula Φ is $(f = 0) \wedge \Phi'$, we say the constraint is* syntactically explicit*.*

Although implicit and explicit ECs have the same logical status, in practice only the syntactically explicit ECs will be known to us and therefore be available to be exploited.

Example 1 *[3] Let f and g be two polynomials,*

1. *The formula $f = 0 \wedge g > 0$ has an explicit EC: $f = 0$.*
2. *The formula $f = 0 \vee g = 0$ has no explicit EC but the equation $fg = 0$ is an implicit EC.*
3. *The formula $f^2 + g^2 \leq 0$ also has no explicit EC, but it has two implicit EC: $f = 0$ and $g = 0$.*
4. *The formula $f = 0 \vee f^2 + g^2 \leq 0$ logically implies $f = 0$, and the equation is an atom of the formula, which makes it an explicit EC according to the definition. As it is not syntactically explicit, since this deduction is semantic rather than syntactic, it is more like an implicit EC in practice.*

In [6] McCallum showed that if the QFF contains an EC, then it is enough to perform a CAD on the variety described by the EC. The remaining constraints must be sign-invariant within the cells where the EC polynomial is zero.

Definition 5 *[6] Let A be a set of polynomial constraints. Let $E \subseteq A$, and define the projection operator $P_E(A) := P(E) \cup \{\mathrm{res}_{x_r}(f, g) \mod f \in E, g \in A \setminus E\}$.*

Theorem 1 *Let $n \geq 2$ and let f, $g \in \mathbb{R}[x_1, \ldots, x_n]$ be real polynomials of positive degrees in the main variable x_n. Let S be a connected subset of $\mathbb{R}^{r-1}$. Suppose that f is delineable on S, in which $R = \mathrm{res}_{x_n}(f, g)$ is order-invariant. Then g is sign-invariant in each section of f over S.*

This projection operator does come with its drawbacks. This is not an inductive algorithm, as the output is a sign-invariant CAD, not order-invariant. Hence this operator can only be used in the first stage of projection. For the subsequent steps, we resort to using the original operator.

3 Lex-least valuation

Recall that *lexicographic order* $\geq_{lex}$ on $\mathbb{N}^n$ is defined by saying that $v = (v_1, \ldots, v_n) \geq_{lex} (w_1, \ldots, w_n) = w$ if and only if either $v = w$ or there exists an $i \leq n$ such that $v_i > w_i$ and $v_k = w_k$ for all k in the range $1 \leq k < i$.

Definition 6 *[10] Let $f \in \mathbb{R}[x_1, \ldots, x_n]$ (with $n \geq 1$) non-zero and $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n$. The lex-least valuation (or Lazard valuation) $\nu_\alpha(f)$ at α is the least (with respect to $\geq_{lex}$) element $v = (v_1, \ldots, v_n) \in \mathbb{N}^n$ such that f expanded about α has the term $c(x - \alpha_1)^{v_1} \cdots (x - \alpha_n)^{v_n}$, where $c \neq 0$.*

Definition 7 *[10] Let $n \geq 2$, take a non-zero element f in $\mathbb{R}[x_1, \ldots, x_n]$ and let $\beta \in \mathbb{R}^{n-1}$. The Lazard residue $f_\beta \in \mathbb{R}[x_n]$ of f at β is defined to be the result of the following algorithm:*

```
$f_\beta \leftarrow f$
for $i \leftarrow 1$ to $n - 1$ do
    $\nu_i \leftarrow$ greatest integer $\nu$ such that $(x_i - \beta_i)^\nu | f_\beta$.
    $f_\beta \leftarrow f_\beta / (x_i - \beta_i)^{\nu_i}$.
    $f_\beta \leftarrow f_\beta(\beta_i, x_{i+1}, \ldots, x_n)$
end for
return $f_\beta$, $(\nu_1, \ldots, \nu_{n-1})$
```

The value of $(\nu_1, \ldots, \nu_{n-1})$ is called the lex-least valuation of f above β.

Definition 8 *[10] Let $S \subseteq \mathbb{R}^{n-1}$ and $f \in \mathbb{R}[x_1, \ldots, x_n]$. We say that f is Lazard delineable on S if:*

i) The lex-least valuation of f above β is the same for each point $\beta \in S$.

ii) There exist finitely many continuous functions $\theta_1 < \ldots < \theta_k$ from $S \to \mathbb{R}$ with $k \geq 0$ such that for all $\beta \in S$, the set of real roots of f_β is $\{\theta_1(\beta), \ldots, \theta_k(\beta)\}$.

iii) If $k = 0$, then the graph of f does not pass over S. If $k \geq 1$, then there exist positive integers $m_1, \ldots, m_k$ such that, for all i and for all $\beta \in S$, m_i is the multiplicity of $\theta_i(\beta)$ as a root of f_β.

Lazard [4] provided the following projection operator (and corresponding changes to lifting) which gave a more efficient CAD, in practice and on average [9].

Definition 9 *[10] Let A be a finite set of irreducible polynomials in $R_n = \mathbb{Z}[x_1, \ldots, x_n]$ with $n \geq 2$. The Lazard projection $PL(A)$ is a subset of R_{n-1} composed of the following polynomials: all leading coefficients of the elements of A; all trailing coefficients of the elements of A; all discriminants of the elements of A; all resultants of pairs of distinct elements of A.*

$PL(A)$ is fairly large if A is large. On the other hand, it is smaller than McCallum's projection operator in [5], as it only asks for the leading and trailing coefficients, not all of them. McCallum et al. [10] gave a validity proof for Lazard's method. We build on [4] and provide a projection operator analogous to the one given in [6]. This projection operator reduces the projection set in the first phase, where there is an EC in the QFF.

Definition 10 *Let A be a set of polynomial constraints. Let $E \subseteq A$, and define the projection operator $PL_E(A) := PL(E) \cup \{\mathrm{res}_{x_n}(f, g) \mid f \in E, g \in A \setminus E\}$.*

Theorem 2 *Let $n \geq 2$ and let f, $g \in \mathbb{R}[x_1, \ldots, x_n]$ be real polynomials of positive degrees in the main variable x_n. Let S be a connected subset of $\mathbb{R}^{r-1}$. Suppose that f is Lazard delineable on S, in which $R = \mathrm{res}_{x_r}(f, g)$ is lex-least invariant. Then g is sign-invariant in each section of f over S.*

4 Conclusions

Our projection operator successfully exploits the presence of one EC but has the drawback that it cannot be used inductively because the output is only a sign-invariant CAD, not order-invariant.

McCallum later provided another projection operator, which does yield an order-invariant CAD.

Definition 11 *[7, 8] Let A be a set of polynomial constraints. Let $E \subseteq A$, and define the projection operator $P_E^*(A) := P_E(A) \cup \{\mathrm{disc}_{x_r}(A \setminus E) \cup \mathrm{coeff}(A \setminus E)\}$, where $\mathrm{coeff}(A \setminus E)$ is the set of all coefficients of polynomials in $A \setminus E$.* [The addition of $\mathrm{coeff}(A \setminus E)$ is due to [8].]

This projection operator deals with multiple equational constraints, reducing the complexity significantly in specific cases. We are currently working on an analogous improvement for the Lazard projection.

References

[1] G.E. Collins. Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition. In *Proceedings 2nd. GI Conference Automata Theory & Formal Languages*, pages 134–183, 1975.

[2] G.E. Collins. Quantifier elimination by cylindrical algebraic decomposition — twenty years of progess. In B.F. Caviness and J.R. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, pages 8–23. Springer Verlag, Wien, 1998.

[3] M. England, R.J. Bradford, and J.H. Davenport. Cylindrical Algebraic Decomposition with Equational Constraints. In J.H. Davenport, M. England, A. Griggio, T. Sturm, and C. Tinelli, editors, *Symbolic Computation and Satisfiability Checking.* Journal of Symbolic Computation (to appear), 2019.

[4] D. Lazard. An Improved Projection Operator for Cylindrical Algebraic Decomposition. In C.L. Bajaj, editor, *Proceedings Algebraic Geometry and its Applications: Collections of Papers from Shreeram S. Abhyankar's 60th Birthday Conference*, pages 467–476, 1994.

[5] S. McCallum. *An Improved Projection Operation for Cylindrical Algebraic Decomposition.* PhD thesis, University of Wisconsin-Madison Computer Science, 1984.

[6] S. McCallum. On Projection in CAD-Based Quantifier Elimination with Equational Constraints. In S. Dooley, editor, *Proceedings ISSAC '99*, pages 145–149, 1999.

[7] S. McCallum. On Propagation of Equational Constraints in CAD-Based Quantifier Elimination. In B. Mourrain, editor, *Proceedings ISSAC 2001*, pages 223–230, 2001.

[8] S. McCallum. Error in [7]. *E-mail 2019 January 5th*, 2019.

[9] S. McCallum and H. Hong. On Lazard's Valuation and CAD Construction. `http://www.arxiv.org/abs/1501.06563`, 2015.

[10] S. McCallum, A. Parusiński, and L. Paunescu. Validity proof of Lazard's method for CAD construction. *J. Symbolic Comp.*, 92:52–69, 2019.

[11] A.S. Nair, J.H. Davenport, and G.K. Sankaran. On Benefits of Equality Constraints in Lex-Least Invariant CAD (Extended Abstract). In *To appear in SC-Square 2019: Satisfiability Checking and Symbolic Computation*, 2019. URL: `https://researchportal.bath.ac.uk/en/publica/on-benefits-of-equality-constraints-in-lex-least-invariant-cad-ex`.