On Benefits of Equality Constraints in Lex-Least Invariant CAD (extended abstract)

Akshar Nair James Davenport Gregory Sankaran

University of Bath, Bath, United Kingdom, BA2 7AY.
{asn42,masjhd,masgks}@bath.ac.uk

Abstract

We consider two methods for CADs, one due to McCallum and one due to Lazard. The former uses order invariant CADs and has been modified, by McCallum himself, so as to take advantage of equational constraints. Lazard's method uses lex-least invariant CADs and is in general faster. In this paper, we start to modify Lazard's method so as to exploit equational constraints. Our algorithm takes in a lex-least invariant CAD of $\mathbb{R}^{n-1}$ as input and outputs a sign invariant CAD of a subvariety of $\mathbb{R}^n$: consequently, it cannot be used recursively, but only for x_n, the first variable to be projected. In the further steps of the projection phase, we use Lazard's original projection operator. Nonetheless, reducing the output in the first step has a domino effect throughout the remaining steps, which significantly reduces the complexity. The long-term goal is to find a general projection operator that takes advantage of the equality constraint and can be used recursively, and this operator gives an important first step in that direction.

1 Introduction

A Cylindrical Algebraic Decomposition (CAD) is a decomposition of a semi-algebraic set $S \subseteq \mathbb{R}^n$ (for any n) into semi-algebraic sets (also known as cells) homeomorphic to $\mathbb{R}^m$, where $1 \leq m \leq n$, such that the projection of any two cells onto the first k coordinates is either the same or disjoint. We generally want the cells to have some property relative to some given set of input polynomials. For example, we might require sign-invariance, i.e. the sign of each input polynomial (often referred to as a constraint) is constant on each cell. Many algorithms for

In: J. Abbott, A. Griggio (eds.): Proceedings of the 4th SC-square Workshop, Bern, Switzerland, 10th July 2019, published at http://ceur-ws.org

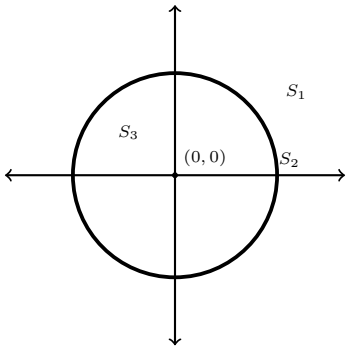
CAD consist of the following three phases.

Projection phase: This phase consists of reducing 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: we choose $x_1 > x_2 > \ldots > x_n$, *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.

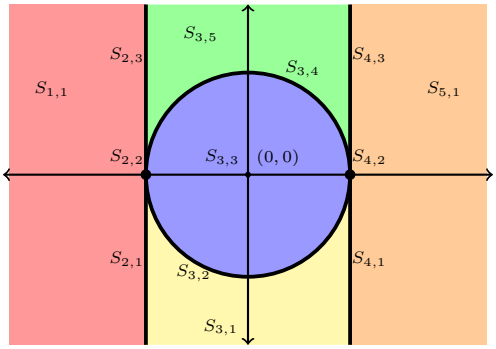
Lifting phase: This phase consists of decomposing higher dimensional spaces using the sample points and the projection polynomials in that dimension, until S is decomposed.

Example 1 *Let us look at the decomposition of* $S = \mathbb{R}^2$ *with respect to the input polynomial:* $x^2 + y^2 - 1$.



$$S_1 = \{x^2 + y^2 - 1 > 0\}; S_2 = \{x^2 + y^2 - 1 = 0\}; S_3 = \{x^2 + y^2 - 1 < 0\}$$

This is not a cylindrical decomposition of $\mathbb{R}^2$*, because when we project* S_1 *and* S_2 *to* $\mathbb{R}^1$ *we get* $\mathbb{R}$ *and* $(-1, 1)$ *respectively. These have non-empty intersection but are not the same. To decompose* S *cylindrically (with ordering* $x > y$*), we take:*



$S_{1,1} = \{x < -1\}, \quad S_{5,1} = \{x > 1\},$
$S_{2,1} = \{x = -1, y < 0\}, \quad S_{2,2} = \{x = -1, y = 0\}, \quad S_{2,3} = \{x = -1, y > 0\},$
$S_{3,1} = \{x^2 + y^2 - 1 > 0 \wedge 1 > x > -1 \wedge y < 0\}, \quad S_{3,2} = \{x^2 + y^2 - 1 = 0 \wedge 1 > x > -1 \wedge y < 0\},$
$S_{3,3} = \{x^2 + y^2 - 1 < 0 \wedge 1 > x > -1\}, \quad S_{3,4} = \{x^2 + y^2 - 1 = 0 \wedge 1 > x > -1 \wedge y > 0\},$
$S_{3,5} = \{x^2 + y^2 - 1 > 0 \wedge 1 > x > -1 \wedge y > 0\},$
$S_{4,1} = \{x = 1, y < 0\}, \quad S_{4,2} = \{x = 1, y = 0\}, \quad S_{4,3} = \{x = 1, y > 0\}.$

The first CAD algorithm, that of Collins [1], gives sign-invariant CADs and has complexity $(2dm)^{3^{2n+O(1)}}$ when the input consists of m polynomials of at most degree d in n variables [2]. A faster algorithm, using order-invariant CADs, was given by McCallum [4] in 1984: it fails, however, if one of the input polynomials is nullified, i.e. vanishes identically above some point in $\mathbb{R}^{n-1}$.

If S is contained in a subvariety it is clearly wasteful to compute a decomposition of $\mathbb{R}^n$. To make this idea more precise, we need some terminology.

Definition 1 *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 $f \in \mathbb{R}[x_1, \ldots, x_n]$ of the form $f \sim 0$ where $\sim \in \{=, <, >\}$ (and by combination also $\{\geq, \leq, \neq\}$).*

Strictly speaking we need only the relation $<$, but this form is more convenient because of the next definition.

Definition 2 *[2] 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 2 *[2] 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, however 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 $f = 0$ an explicit EC according to the definition. However, it is not a syntactically explicit EC. As the logical deduction is semantic rather than syntactic, the EC is unlikely to be picked up by a basic input parser and so has more in common with an implicit EC.*

Definition 3 *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 the cells of a decomposition, we say the decomposition is P-invariant.* Common P are sign [1] and order [4].

ECs were first exploited in 1999 by McCallum in [5], which modifies the process of [4] so as to construct a decomposition of the variety described by the EC rather than a decomposition of $\mathbb{R}^n$. This method lifts an order-invariant CAD of $\mathbb{R}^{n-1}$ to a sign-invariant CAD of some variety in $\mathbb{R}^n$.

Because the output is not order-invariant the algorithm is not recursive. Later, McCallum [6] gave a modified algorithm that does output an order-invariant CAD. It can be used recursively to deal with multiple equational constraints, which reduces the complexity significantly in specific cases.

In 1994, Lazard [3] provided a different approach to constructing a CAD of $\mathbb{R}^n$, in which the order of a polynomial is replaced by the lex-least valuation (see Definition 4 below). Lazard's approach allows a slightly smaller projection operator, because unlike McCallum's, it does not require middle coefficients, only leading and trailing ones: also the lifting process is significantly different. It also handles the cases in which McCallum's algorithms fail due to nullification, when a polynomial vanishes identically.

In this paper, we modify Lazard's process so as to exploit a single equational constraint. The result is similar to the projection operator of [5]. It works well for syntactically explicit ECs: in other cases, we would resort to using the normal projection operator. Like [5], it cannot be used inductively, since it outputs a sign-invariant CAD rather than a lex-least valuation-invariant CAD.

Our improvement is that we only need to compute Lazard's projection operator on the equational constraints. The only additional polynomials are the resultants between the equational and non-equational constraints. This is a considerable improvement over Collins' projection operator and McCallum's modifications, as the projection set in our case is significantly smaller. This will be further discussed in Section 4.

2 Lex-least Valuation

In order to understand lex-least valuation, let us recall *lexicographic order* $\geq_{lex}$ on $\mathbb{N}^n$, where $n \geq 1$. We say 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 4 *[8, Definition 2.4] Let $n \geq 1$ and suppose that $f \in \mathbb{R}[x_1, \ldots, x_n]$ is non-zero and $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n$. The lex-least 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*

$$c(x_1 - \alpha_1)^{v_1} \cdots (x_n - \alpha_n)^{v_n},$$

where $c \neq 0$.

Note that $\nu_\alpha(f) = (0, \ldots, 0)$ if and only if $f(\alpha) \neq 0$. The lex-least valuation is referred to as the Lazard valuation in [8]. Later we shall also use the Lazard valuation, and some facts about it, for formal power series rather than polynomials. To justify this requires some technical detail which we omit here.

Example 3 *If $n = 1$ and $f(x_1) = x_1^3 - 2x_1^2 + x_1$, then $\nu_0(f) = 1$ and $\nu_1(f) = 2$. If $n = 2$ and $f(x_1, x_2) = x_1(x_2 - 1)^2$, then $\nu_{(0,0)}(f) = (1, 0)$, $\nu_{(2,1)}(f) = (0, 2)$ and $\nu_{(0,1)}(f) = (1, 2)$.*

The following are the important properties of the lex-least valuation, taken from [8, Section 3].

Proposition 1 (Valuation) *ν_α is a valuation: that is, if f and g are non-zero elements of $\mathbb{R}[x_1, \ldots, x_n]$ and $\alpha \in \mathbb{R}^n$, then $\nu_\alpha(fg) = \nu_\alpha(f) + \nu_\alpha(g)$ and $\nu_\alpha(f + g) \geq_{lex} \min\{\nu_\alpha(f), \nu_\alpha(g)\}$.*

Proposition 2 (Upper semicontinuity) *Let f be a nonzero element of $\mathbb{R}[x_1, \ldots, x_n]$ and let $a \in \mathbb{R}^n$. Then there exists an open neighbourhood $V \subset \mathbb{R}^n$ of a, such that $\nu_b(f) \leq_{lex} \nu_a(f)$ for all $b \in V$.*

Proposition 3 *Let f and g be non-zero elements of $\mathbb{R}[x_1, \ldots, x_n]$ and let $S \subset \mathbb{R}^n$ be connected. Then fg is lex-least invariant in S if and only if f and g are lex-least invariant in S.*

Proof. Suppose that fg is lex-least invariant, say $v_b(fg) = v$ for all $b \in S$, so $v_b(g) = v - v_b(f)$. Choose $a \in S$ then by Proposition 2 the set $S_+ = \{b \in S \mid v_b(f) \leq_{lex} v_a(f)\}$ is open. But

$$S_- = \{b \in S \mid v_b(f) \geq_{lex} v_a(f)\} = \{b \in S \mid v - v_b(g) \geq_{lex} v - v_a(g)\} = \{b \in S \mid v_b(g) \leq_{lex} v_a(g)\}$$

is also open. Hence $S_+ \cap S_- = \{b \in S \mid v_b(f) = v_a(f)\}$ is open, for any $a \in S$, so the function $b \mapsto v_b(f)$ is a continuous function from S to $\mathbb{Z}^n$. As S is connected, this function is constant, i.e. f is lex-least invariant. The other implication is trivial. □

Definition 5 *[8] 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, and the lex-least valuation of f above β are defined to be the result of Algorithm 1:*

Algorithm 1 Lazard residue

1: $f_\beta \leftarrow f$
2: **for** $i \leftarrow 1$ to $n-1$ **do**
3: $\quad \nu_i \leftarrow$ greatest integer ν such that $(x_i - \beta_i)^\nu | f_\beta$.
4: $\quad f_\beta \leftarrow f_\beta / (x_i - \beta_i)^{\nu_i}$.
5: $\quad f_\beta \leftarrow f_\beta(\beta_i, x_{i+1}, \ldots, x_n)$
6: **end for**
7: **return** f_β, $(\nu_1, \ldots, \nu_{n-1})$

The value of $(\nu_1, \ldots, \nu_{n-1}) \in \mathbb{Z}^{n-1}$ is called the lex-least valuation of f above $\beta \in \mathbb{R}^{n-1}$ (not to be confused with the lex-least valuation at $\alpha \in \mathbb{R}^n$, defined in Definition 4). We denote it by $N_\beta(f)$. Notice that if $b = (\beta, b_n) \in \mathbb{R}^n$ then $\nu_b(f) = (N_\beta(f), \nu_n)$ for some integer ν_n: in other words, $N_\beta(f)$ consists of the first $n-1$ coordinates of the valuation of f at any point above β.

Definition 6 *[8, definition 2.10] 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 $\theta_i \colon S \to \mathbb{R}$, such that $\theta_1 < \ldots < \theta_k$ if $k > 0$ and 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 $\beta \in S$ and for all $1 \leq i \leq k$, m_i is the multiplicity of $\theta_i(\beta)$ as a root of f_β.

Definition 7 *[8, Definition 2.10] Let f be Lazard delineable on $S \subseteq \mathbb{R}^{n-1}$. Then*

i) The graphs θ_i are called Lazard sections and m_i is the associated multiplicity of these sections.

ii) The regions between consecutive Lazard sections[1] are called Lazard sectors.

[1] Including $\theta_0 = -\infty$ and $\theta_{k+1} = +\infty$.

For later use, we propose some geometric terminology to describe the conditions under which a polynomial is nullified in the terminology of [5].

Definition 8 *A variety $C \subseteq \mathbb{R}^n$ is called a curtain if, whenever $(\underline{x}, x_n) \in C$, then $(\underline{x}, y) \in C$ for all $y \in \mathbb{R}$.*

Definition 9 *Suppose $f \in \mathbb{R}[x_1, \ldots, x_n]$ and $W \subseteq \mathbb{R}^{n-1}$. We say that f has a curtain at W if for all $\underline{x} \in W$ and $y \in \mathbb{R}$ we have $f(\underline{x}, y) = 0$.*

Remark 1 *Lazard delineability differs from delineability as in [1] and [5] in two important ways. First, we require lex-least invariance on the sections. Second, delineability is not defined on curtains, but Lazard delineability is because the Lazard sections are of f_β rather than f.*

Proposition 4 *[8, Proposition 2.11] Let $f \in \mathbb{R}[x_1, \ldots, x_n]$ be of positive degree in x_n and let S be a connected subset of $\mathbb{R}^{n-1}$. Suppose that f is Lazard delineable on S. Then f is lex-least invariant in each Lazard section and sector of f over S.*

Proof. We know that $N_\beta(f)$, the lex-least valuation of f above β, for all $\beta \in S$, is invariant, meaning that for $\alpha \in \mathbb{R}^n$ in the sections of f over S, the first $n-1$ coordinates of $\nu_\alpha(f)$ agree with $N_\beta(f)$. Consider a section given by θ with associated multiplicity m, and $\alpha = (\beta, \theta(\beta))$ the point of this section above β. Then the last coordinate of $\nu_\alpha(f)$ is m, and hence f is lex-least invariant in every section of f over S. It is also invariant on sectors because there the last component of the valuation is just 0. □

Remark 2 *We can use Algorithm 1 to compute the lex-least valuation of f at $\alpha \in \mathbb{R}^n$. After the loop is finished, we proceed to the first step of the loop and perform it for $i = n$ and the n-tuple $\nu_1, \ldots, \nu_n$ is the required valuation.*

3 Lex-least Invariance

This section contains details of Lazard's original projection operator used to obtain a lex-least invariant CAD.

Definition 10 *[8, definition 2.1] Let A be a finite set of irreducible polynomials in $\mathbb{R}[x_1, \ldots, x_n]$ with $n \geq 2$. The Lazard projection $\mathrm{PL}(A)$ is a subset of $\mathbb{R}[x_1, \ldots, x_{n-1}]$ composed of the following polynomials.*

i) All leading coefficients of the elements of A.

ii) All trailing coefficients of the elements of A.

iii) All discriminants of the elements of A.

iv) All resultants of pairs of distinct elements of A.

Experimentally this projection operator gives a more efficient CAD, in practice and on average [7]. This projection operator is fairly large in the event that we have a large number of polynomials in A. On the other hand, it is smaller than McCallum's projection operator in [4], as it only asks for the leading and trailing coefficients, rather than all of them.

McCallum et al. [8] gave a proof for the following theorem. It underpins the use of Lazard's projection operator in producing a lex-least invariant CAD.

Theorem 1 *[8, Theorem 5.1] Let $f \in \mathbb{R}[x_1, \ldots, x_n]$ be of positive degree in x_n. Let D, l and t denote the discriminant, leading coefficient and trailing coefficient (i.e. the coefficient of x_n^0) of f respectively, and suppose that each of these polynomials is not identically zero (as elements of $\mathbb{R}[\underline{x}]$). Let S be a connected submanifold of $\mathbb{R}^{n-1}$ in which D, l and t are all lex-least invariant. Then f is Lazard delineable on S, and this implies that f is lex-least invariant in every Lazard section and sector over S.*

In summary, Theorem 1 says that if $\mathrm{PL}(A)$ is lex-least invariant over a section S, then all polynomials in A are Lazard delineable over S. This implies that the projection operator can be used inductively, and this makes it possible to provide a lex-least invariant CAD.

4 Modification to Lazard's Projection Operator

This section focuses on the main claim of this paper. We first define our modification of Lazard's projection operator. This projection operator reduces the projection set in the first phase, where there is an EC in the QFF. This is similar to McCallum's modification in [5] of his original operator in [4]. The reason for considering the resultants here is the same: we are performing the P-invariant CAD on the hypersurface $f = 0$, rather than $\mathbb{R}^n$.

Definition 11 *Let $A \subset \mathbb{R}[x_1, \ldots, x_n]$ be a set of polynomial constraints. Let $E \subseteq A$, and define the projection operator $\mathrm{PL}_E(A)$ as follows:*

$$\mathrm{PL}_E(A) = \mathrm{PL}(E) \cup \{\mathrm{res}_{x_n}(f, g) \mid f \in E,\, g \in A \setminus E\}.$$

Remark 3 *In practice we will choose E to consist of polynomials occurring in equational constraints.*

Theorem 2 *Let $n \geq 2$ and let $f, g \in \mathbb{R}[x_1, \ldots, x_n]$ be of positive degrees in the main variable x_n. Let S be a connected subset of $\mathbb{R}^{n-1}$. Suppose that f is Lazard delineable on a connected submanifold $S \subset \mathbb{R}^{n-1}$, in which $R = \mathrm{res}_{x_n}(f, g)$ is lex-least invariant and that f does not have a curtain on S. Then g is sign-invariant in each section of f over S.*

Proof. Let σ be a section of f over S, given by the map $\theta\colon S \to \mathbb{R}$. Since g is a continuous function it is enough to show that g is sign-invariant in a euclidean neighbourhood in σ of an arbitrary point $\alpha \in \sigma$. We may take α to be the origin, and we may also assume that $g(\alpha) = 0$, since otherwise g is sign-invariant in a neighbourhood of α in $\mathbb{R}^n$, and in particular in a neighbourhood in σ.

Now, by the definition of θ, we have $\theta(0, \ldots, 0) = 0$ and also $f(0, \ldots, 0) = 0$. Suppose that $x_n = 0$ has multiplicity $m \neq 0$ as a root of $f(0, \ldots, 0, x_n) = 0$. This implies that $f(0, \ldots, 0, x_n) = \bar{q}(x_n) \cdot x_n^m$ where $\bar{q}(0) \neq 0$. Therefore, by Hensel's Lemma (as in [5]), there exists a euclidean neighbourhood N_1 of the origin in $\mathbb{R}^n$ and formal power series $q(x_1, \ldots, x_n)$ and $h(x_1, \ldots, x_n)$ such that $f(x_1, \ldots, x_n) = q(x_1, \ldots, x_n) \cdot h(x_1, \ldots, x_n)$, where $q(0, \ldots, 0, x_n) = \bar{q}(x_n)$ and $h(0, \ldots, 0, x_n) = x_n^m$.

Since $q(0, \ldots, 0) \neq 0$ in N_1, there exist $\epsilon > 0$ and an open neighbourhood N_2 of the origin in $\mathbb{R}^{n-1}$, such that $q(x_1, \ldots, x_n) \neq 0$ for all $(x_1, \ldots, x_n) \in N_2 \times (-\epsilon, \epsilon)$. Since θ is continuous, we can shrink N_2 to get $\theta(x_1, \ldots, x_{n-1}) \in (-\epsilon, \epsilon)$ for all $(x_1, \ldots, x_{r-1}) \in N_2$. In essence the subsection of σ (of f over N_2) lies in $N_2 \times (-\epsilon, \epsilon)$. If $\alpha' \in S \cap N_2$ then, since f is Lazard delineable, $\nu_{(\alpha', \theta(\alpha'))}(f) = \nu_\alpha(f)$.

On the other hand, $\nu_\alpha(h) = (0, \ldots, m)$. Since $f = q \cdot h$ and f is Lazard delineable in $S \cap N_2$, this implies that $\nu_{(\alpha', \theta(\alpha'))}(h) = (0, \ldots, m)$.

Denote by P the resultant of h and g with respect to x_n. As in the proof of [5, Theorem 2.2] we have $R = Q \cdot P$ for some suitable formal power series Q in the first $n-1$ variables. We also have $P = 0$ at α since h and g both vanish there, so $\nu_\alpha(P)$ is non-zero. By assumption R is lex-least invariant in the region N_2, so P is lex-least invariant in N_2 (by Proposition 3).

Since $h = 0$ in $\sigma \cap (N_2 \times (-\epsilon, \epsilon)) \cap \sigma$ and P is lex-least invariant in N_2, we see that $g = 0$ in $\sigma \cap (N_2 \times (-\epsilon, \epsilon))$ which is a euclidean neighbourhood of α in σ. Thus g is sign-invariant in σ, which implies that it is sign-invariant in S. □

Theorem 3 *Let $A \subset \mathbb{R}[x_1, \ldots, x_n]$ be a set of irreducible polynomials in n variables of positive degrees in x_n, where $n \geq 2$. Let E be a subset of A. Let S be a connected submanifold of $\mathbb{R}^{n-1}$. Suppose that each element of $\mathrm{PL}_E(A)$ is lex-least invariant in S. Then each element of E either vanishes identically above S (i.e. has a curtain) or is Lazard delineable on S. The sections over S of the elements of E which do not vanish identically on S are pairwise disjoint, each element of E is lex-least invariant in every such section, and each element of A not in E is sign-invariant in every such section.*

Proof.
This follows directly from Theorem 1 and Theorem 2. □

Theorem 3 does come with its flaws, similar to McCallum's single equational constraint [5]. The method cannot be used recursively as it will output a sign-invariant CAD.

5 Conclusion and Further Research

The main requirement for Theorem 3 to work is that the Equational Constraint in the input QFF must not have curtains. This is superficially similar to the "no nullification" requirements of [4, 5, 6], but is less constraining in two ways.

1. It only applies to the EC, not to the other polynomials involved.

2. Because we ae using Lazard lifting, rather than that of [4], this constraint only applies to x_n, and not recursively.

The other limiting factor for this projection is that it cannot be used recursively because it outputs a sign-invariant CAD, rather than a lex-least invariant CAD. This limits us to using the modified projection operator $\mathrm{PL}_E(A)$ only in the first step of the projection phase, and Lazard's projection operator $\mathrm{PL}(A)$ for the subsequent steps. Despite these drawbacks, reducing the output of the first step in the projection phase has a domino effect on further steps.

McCallum's projection operator in [6] shows that it is possible to benefit from having an EC in the QFF, and provides an order-invariant CAD. By contrast the projection operator in [5] only provides a sign-invariant CAD. Similarly, we hope to find a projection operator that outputs a lex-least invariant CAD so that it can be used recursively. This modification would reduce the complexity of the algorithm significantly.

References

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

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

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

[4] McCallum, S.: An Improved Projection Operation for Cylindrical Algebraic Decomposition. Ph.D. thesis, University of Wisconsin-Madison Computer Science (1984)

[5] McCallum, S.: On Projection in CAD-Based Quantifier Elimination with Equational Constraints. In: Dooley, S. (ed.) Proceedings ISSAC '99. pp. 145–149 (1999)

[6] McCallum, S.: On Propagation of Equational Constraints in CAD-Based Quantifier Elimination. In: Mourrain, B. (ed.) Proceedings ISSAC 2001. pp. 223–230 (2001)

[7] McCallum, S., Hong, H.: On Using Lazard's Projection in CAD Construction. J. Symbolic Computation **72**, 65–81 (2016)

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