Theory of Hybrid Systems Informatik 2
RWTH AACHEN UNIVERSITY

**The present work was submitted to the LuFG Theory of Hybrid Systems**

BACHELOR OF SCIENCE THESIS

# COMPARING DIFFERENT PROJECTION OPERATORS IN THE CYLINDIRCAL ALGEBRAIC DECOMPOSITION FOR SMT SOLVING

**Tarik Viehmann**

*Examiners:*
Prof. Dr. Erika Ábrahám
PD Dr. Viktor Levandovskyy

Aachen, 30. September 2016

**Abstract**

One task in the field of satisfiability modulo theories (SMT) is to solve formulas that are build out of nonlinear real arithmetic constraints. A technique suitable in this context is the cylindrical algebraic decomposition procedure (CAD). Given constraints consisting of $\tau$-variate polynomials, CAD is used to create sample points of the $\mathbb{R}^\tau$ space, that are sufficient to deduce the satisfiability of these constraints. Core of the CAD procedure is to reduce the problem setting to the case of univariate polynomials by projecting the input polynomials down. There are various improvements made on the field of designing suiting projection operators. This paper wraps up the design of Collins first operator, Hongs refinement, McCallums approach and its refinement by Brown. Then the operators get tested by implementing them to the SMT-RAT framework. The theoretical improvements are reflected in the benchmarks and the quality of improvement is reviewed.

# Erklärung

Hiermit versichere ich, dass ich die vorgelegte Arbeit selbstständig verfasst und noch nicht anderweitig zu Prüfungszwecken vorgelegt habe. Alle benutzten Quellen und Hilfsmittel sind angegeben, wörtliche und sinngemäße Zitate wurden als solche gekennzeichnet.

Tarik Vihemann
Aachen, den 30. September 2016

# Contents

# Chapter 1

# Introduction

The research field of satisfiability modulo theories (SMT) deals with decision procedures of first-order logic formulas, where the predicates, functions and constants are fixed by some background theories. In this paper the theory of quantifier-free nonlinear real arithmetics (QF_NRA) is of interest. This theory allows to add atoms of the form $f < g$ to the boolean structure of a formula, where $f$ and $g$ are multivariate polynomials with integer coefficients. Figure 1.1 shows the standard syntax of NRA formulas. The quantifier-free fragment does not allow the negation of formulas that contain existential quantifiers as this would require additional quantifier elimination methods.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Polynomials | $t$ | ::= | $0$ | $\mid$ | $1$ | $\mid$ | $x$ | $\mid$ | $t + t$ | $\mid$ $t * t$ |
| Constraints | $c$ | ::= | $t < t$ | | | | | | | |
| Formulas | $\varphi$ | ::= | $c$ | $\mid$ | $\neg\varphi$ | $\mid$ | $\varphi \vee \varphi$ | $\mid$ | $\exists x.\varphi$ | |

where $x$ is a variable of domain $\mathbb{R}$

Figure 1.1: Syntax of NRA formulas

The task of solving QF_NRA formula is known to be decidable as it was proven by Tarski [Tar48] and of great interest in both industrial and scientific context, because of the expressive power, that QF_NRA formulas provide to model real world problem instances.
Given such a QF_NRA formula $\varphi$, a SAT solver can check for satisfiability of the boolean skeleton of $\varphi$, dealing also with the existential quantifiers. If $\varphi$ is already unsatisfiable by its boolean structure, it can be returned, else the SAT solver finds an assignment for the boolean skeleton, which translates into a set of constraints $c_1, \ldots, c_m$ that must hold in order to satisfy $\varphi$.
The cylindrical algebraic decomposition (CAD) procedure can be used as a method, to check whether such a set of constraints can be satisfied or not. It uses the fact, that a constraint $c_j$ of the form $c_j := g < h$ can be transformed to $f < 0$ with $f := g - h$. So the task can be simplified as it is sufficient to analyze a given set $A_\tau = \{f_1, \ldots, f_m\}$ of $\tau$-variate polynomials, $\tau \in \mathbb{N}$, with respect to the arrangement of their real roots. As the domain for the variables is $\mathbb{R}^\tau$, the CAD procedure partitions the $\mathbb{R}^\tau$ space into finitely many nonempty connected subsets $D_1, \ldots, D_n$ (called regions). These regions

are arranged in a special way, that is called "cylindrical". Each region is constructed, so that $A_\tau$ is sign-invariant on each subset $D_i$.

**Definition 1.0.1.** *Let $X$ be a subset of $\mathbb{R}^\tau$ and let $f \in \mathbb{Z}[x_1,\ldots,x_\tau]$. $f$ is sign-invariant on $X$, if one of the following conditions holds:*

- $f(x) > 0 \forall x \in X$
- $f(x) = 0 \forall x \in X$
- $f(x) < 0 \forall x \in X$

*Let $A$ be a finite subset of $\mathbb{Z}[x_1,\ldots,x_\tau]$. $A$ is sign-invariant on $X$, if each $f \in A$ is sign-invariant on $X$.*
*Let $D = \{X_1,\ldots,X_n\}$ be a set of disjoint subsets of $E^\tau$. $A$ is sign-invariant on $D$, if $A$ is sign-invariant on $X_i, \forall 1 \leq i \leq n$.*

The regions $D_i, 1 \leq i \leq n$ are therefore chosen in a way, that the satisfiability of each constraint $c_j, 1 \leq j \leq m$ does not change, independent from the sample point $\alpha \in D_i$ that could be taken, to check. This justifies to only take one sample point out of each region and test, whether one of them satisfies all constraints, or not. If such a sample point exists, "satisfiable" is returned, else "unsatisfiable".
During the CAD procedure, the problem of finding those regions for the $\mathbb{R}^\tau$ space is reduced to that of finding regions of $\mathbb{R}$. This is done by the use of a projection operator, that relaxes the problem setting by inductively eliminating one dimension. After regions of $R$ are determined, sample points are chosen and then lifted up again to sample points of $\mathbb{R}^\tau$ that represent the regions $D_1,\ldots,D_n$. The projection operator is of great interest, as it has direct influence on the number of sample points, that are constructed in the end.

This thesis gives an overview about the theoretical progress that was made with respect to the projection operator within the CAD procedure, since Collins first came up with CAD in the context of quantifier elimination in [Col75]. His work got presented in [ACM84], where a full CAD algorithm is proposed. The other work covered in this thesis are the refinements of Hong in [Hon90], McCallum in [McC85] and Brown in [Bro01a]. It aims at stating out the core ideas for the operators as well as pointing out the changes, that come along by using the different approaches. Furthermore the four projection operators got implemented in the SMT-RAT framework and analyzed with respect to the growth of polynomials in each projection step, their degree and the time to establish a full CAD on the benchmarks of QF_NRA formulas of the SMT-LIB.
Chapter 2 gives an abstract overview over the CAD procedure as well as the exact definition of a CAD. Then in chapter 3 some mathematical background knowledge is collected, mostly on the field of ring theory and results in the theory of resultants. The presented theorems are used to give an inside on the validation of the presented operators, which get formulated in the following chapter. In section 5.1 the proof sketch for Collins operator is depicted. Then Hongs justification for his improved operator is given, followed by a section dedicated to McCallums work. The focus there is to explain, why his operator underlays some restrictions, like the mandatory usage of finest square-free bases and the restriction to well-oriented polynomials, and what needs to be noted, when using his operator in the CAD procedure. In section 5.4 Browns improvement on McCallums operator is explained and the important

changes between them are highlighted. Then the practical results of the benchmarks are presented and reviewed.

# Chapter 2

# CAD Procedure

## 2.1 The Role of Projection Operators in CAD

The motivation for the CAD algorithms, that were developed in the past decades, is, that creating a CAD is easy in case of univariate polynomials. A CAD is determined by the arrangement of the real roots of each polynomial in the input set. It is common knowledge that zeros of univariate polynomials can be computed efficiently. The CAD in the univariate case therefore consists of all real roots of the polynomials, the open intervals between the real roots and the open intervals from the smallest zero to $-\infty$ and from the greatest real root to $\infty$. Figure 2.1 shows decomposition of $\mathbb{R}$ into a $\{f,g\}$-invariant CAD $\{R_1,\ldots,R_{11}\}$.

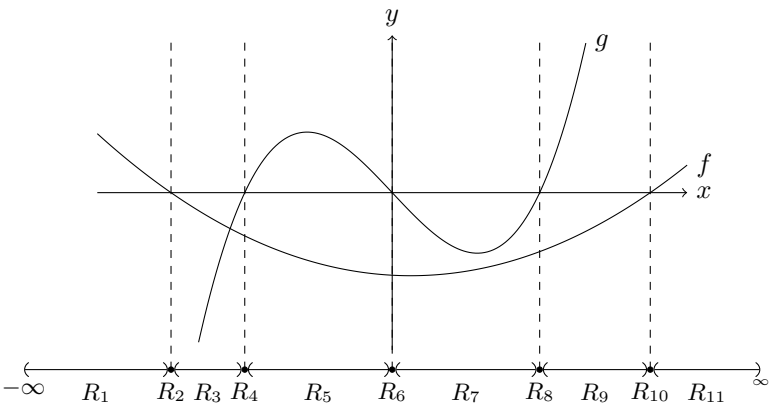


Figure 2.1: $\{f,g\}$-invariant CAD of $\mathbb{R}$

The idea in CAD is, that the $\tau$-variate polynomials are projected to univariate polynomials by eliminating one dimension in each projection step. Then a CAD of the $\mathbb{R}$ gets calculated and inductively lifted to a CAD of $\mathbb{R}^\tau$. Therefore CAD algorithms work in 3 phases: projection, base and extension phase. In the projection phase, $A_\tau$ is projected to $A_{\tau-1}$ (a set of $(\tau-1)$-variate polynomials) with the use of a projection operator *proj*. *proj* is designed in a way, that any on $proj(A_i)$ sign-invariant CAD can be extended to a sign-invariant CAD on $A_i$. This process is repeated inductively

until $A_1$ is constructed. In the base phase a CAD of $\mathbb{R}^1$ is constructed out of $A_1$ by finding the real roots of all polynomials and choosing a sample point inside of each open interval on $\mathbb{R}$ between two zeros (or $-\infty/\infty$). Thus it yields a list of sample points $p_1, \ldots, p_t \in \mathbb{R}$. The lifting phase then lifts the sample points in $\mathbb{R}^i$ to sets of sample points in $\mathbb{R}^{i+1}$ inductively until a CAD of $\mathbb{R}^\tau$ is produced.
One bottleneck during CAD procedure turns out to be an exponential growth in the size of the projected polynomials due to the inductive usage of *proj*. This carries on to an inductive growth on the number of sample points due to the inductive lifting with respect to the projected polynomials. This doubly-exponential growth that is directly influenced by the quick growth in polynomials during projection in each step is the reason why this paper focuses on observing the quality of improvements that were made since Collins et al. proposed the first complete CAD algorithm in [ACM84] based on his work on quantifier elimination methods [Col75]. The improvements covered are: The refinement of Collins operator by Hong [Hon90], the reduced operator under the assumption of well-oriented polynomials by McCallum [McC85] and the improvement on McCallums operator by Brown [Bro01a] with the help of a modified lifting phase.

## 2.2 Definition of CAD

Before getting down to the different operators, the exact definition of a cylindrical algebraic decomposition is given here as compact as possible. The naming conventions are taken from [ACM84].

**Definition 2.2.1.** *For $M \subseteq \mathbb{R}^\tau, M \neq \emptyset$, a finite set of disjoint regions, whose union is $M$ is called **decomposition** of $M$.*

To define the term "cylindrical", some more notations are needed:

**Definition 2.2.2.**
1. *For a region $R$, the **cylinder** over $R$ is defined as $Z(R) := R \times \mathbb{R}$.*
2. *The graph of a a continuous function $f : R \to \mathbb{R}$ defines a **section** of $Z(R)$.*
3. *Let $f_1, f_2 : R \to \mathbb{R}$ be continuous functions with $f_1 < f_2$. The set of points $\{(\alpha,\beta) \in Z(R) \mid \alpha \in R, \beta \in \mathbb{R}, f_1(\alpha) < \beta < f_2(\alpha)\}$ is called $(f_1,f_2)$-**sector** of $Z(R)$.*

Note that a finite set of continuous functions $f_1 < f_2 < \ldots < f_k, k \geq 0$ defined on $R$ induce a decomposition of $Z(R)$ by the $(f_i,f_{i+1})$-sectors for $0 \leq i \leq k$ and $f_0 := -\infty, f_{k+1} := \infty$ together with the $f_i$-sections of $Z(R)$ for $1 \leq i \leq k$.

**Definition 2.2.3.** *A decomposition of $Z(R)$, that is induced by $f_1 < f_2 < \ldots < f_k, k \geq 0$ is called a **stack over** $R$.*

This is sufficient to formulate the cylindrical property that a CAD has.

**Definition 2.2.4.** *A decomposition $D$ of $\mathbb{R}^1$ is **cylindrical**, if $D$ is a stack over $\mathbb{R}^0$ (a single point).*
*A decomposition $D$ of $\mathbb{R}^\tau, \tau > 1$ is **cylindrical**, if there is a decomposition $D'$ of $\mathbb{R}^\tau$, such that for any region $R$ there is a subset of $D$ that is a stack over $R$.*

Now the definition of an algebraic decomposition can be given:

**Definition 2.2.5.** *A set $R \subset \mathbb{R}^\tau$ is **semi-algebraic**, if there is a NRA formula $\varphi$, which defines $R$:*
$\varphi(x_1,\ldots,x_\tau) = TRUE \Leftrightarrow (x_1 \ldots,x_\tau) \in \mathbb{R}^\tau$

Note that this is not the general definition of semi-algebraic sets, but only a sufficient one, when dealing with subsets of $\mathbb{R}^\tau$. See [ACM84] chapter 2 for more details on this.

**Definition 2.2.6.** *A decomposition $D$ of $\mathbb{R}^\tau$ is **algebraic**, if each region $R \in D$ is a semi-algebraic set.*

# Chapter 3

# Results of Algebra and the Theory of Resultants

This chapter gives a rough overview over the theorems and notations that are used within this paper. To begin with, here $0 \in \mathbb{N}$. Since the algebraic structure of the polynomials is $\mathbb{Z}[x_1, \ldots, x_\tau]$, some properties of this ring is given here.
*char* denotes the characteristic of a ring. Let $R$ be a unitary ring, $char(R) := n$, if $n$ is the smallest positive integer such that $\sum_{i=1}^{n} 1 = 0$ in $R$. $char(R) := 0$ if no such integer exists. So $char(\mathbb{Z}[x_1, \ldots, x_\tau]) = 0$ and $char(\mathbb{R}) = 0$.
$\mathbb{Z}$ is a unique factorization domain (UFD), since the unique (up to ordering) prime factorization of $\mathbb{N}$ can be extended to an unique (up to ordering and multiplication with the units $\mathbb{Z}^* := \{1, -1\}$ of $\mathbb{Z}$) factorization.
As Gauss's theorem in ring theory states, $\mathbb{Z}[x]$ is also a UFD.

**Theorem 3.0.1** (Gauss - Ring Theory)**.** *Let $R$ be a UFD $\Rightarrow R[x]$ is a UFD.*

*Proof reference.* See theorem 18.29 [Jud16]. □

The definition for multivariate polynomial rings implies that $\mathbb{Z}[x_1, \ldots, x_\tau]$ also is a UFD.

**Definition 3.0.2.** *Let $R$ be a commutative unitary ring, $\{x_1, \ldots, x_\tau\}$ a set of variables, $R[x_1, \ldots, x_\tau] := S[x_\tau]$, with $S := R[x_1, \ldots, x_{\tau-1}]$ (inductively).*

This also justifies a different view on multivariate polynomials: Assuming a variable order of $x_1 < \ldots < x_\tau$ one could look at $x_\tau$ as the main variable of any polynomial $f \in \mathbb{Z}[x_1, \ldots, x_\tau]$ which has coefficients $a_i \in \mathbb{Z}[x_1, \ldots, x_{\tau-1}]$ with $f = \sum_{i=0}^{n} a_i x^i$ for some $n \in \mathbb{N}$.
Depending on the chosen main variable, the primitive part of a polynomial can be defined as well as the content, the leading coefficient, reductum and its degree. Note that within this paper the variable order stays fixed and the lexicographically greatest variable is set to be the main variable. So the ring $\mathbb{Z}[x_1, \ldots, x_\tau]$ is interpreted as $R[x_\tau] = \mathbb{Z}[x_1, \ldots, x_\tau]$ with $R := \mathbb{Z}[x_1, \ldots, x_{\tau-1}]$. Also GCD denotes the greatest common divisor function.

**Definition 3.0.3.** *Let $R$ be be a unique factorization domain, $f \in R[x]$. Let $n \in \mathbb{N}$ such that $f = \sum_{i=0}^{n} a_i x^i$, $a_i \in R$,$0 \leq i \leq n$ and with $a_n \neq 0$.*

$$\begin{aligned} cont_x(f) &:= GCD(a_0, \ldots, a_n) \\ prim_x(f) &:= f/cont_x(f) \\ deg_x(f) &:= n \\ ldcf_x(f) &:= a_n \\ red_x(f) &:= f - a_n x^n \end{aligned}$$

Since the variable order is obvious in this paper, the subscript may be omitted. Also points of the form $(\alpha_1, \ldots, \alpha_\tau, \beta)$, where $\alpha = (\alpha_1, \ldots, \alpha_\tau) \in \mathbb{R}^\tau$ and $\beta \in \mathbb{R}$ may be abbreviated to $(\alpha, \beta)$.
Another required concept is the use of finest square-free bases, which needs the definition of square-free factorization.

**Definition 3.0.4.** *Let $R$ be a UFD, $0 \neq f \in R[x]$. $f$ is square-free, if $g^2$ does not divide $f, \forall g \in R[x] \setminus R$. There exists $f_1, \ldots, f_k \in R[x] \setminus \{0\}$ with $k > 0$ and*

- *$f_i$ is square-free $\forall 1 \leq i \leq k$,*
- *$f_i = prim_x(f_i), \forall 2 \leq i \leq k$,*
- *$GCD(f_i, f_j) = 1, \forall 1 \leq i < j \leq k$ and $f = \prod_{i=1}^{k} f_i^i$.*

*The $f_i$ are unique up to factors of units $R^*$ of $R$. This factorization is called square-free factorization. $f_1, \ldots, f_k$ are called the square-free factors.*

The finest square-free base of a set of polynomials

$$P := \{p_1, \ldots, p_n\} \subset \mathbb{Z}[x_1, \ldots, x_\tau]$$

denotes the set of all irreducible factors of $\prod_{i=1}^{k} p_i$. This set then contains only factors that are square-free and pairwise relatively prime.
Another useful fact comes from the observation that the GCD of a polynomial and its derivative contains all square-free factors $f_i$ with multiplicity $i - 1$.

**Theorem 3.0.5.** *Let $R$ be a unique factorization domain with $char(R) = 0$, $f \in R[x]$ with square-free factorization $f = \prod_{i=1}^{k} f_i^i$. Then the following equation holds:*

$$gcd(f, f') = cont(f) \cdot \prod_{i=2}^{k} f_i^{i-1}$$

*Proof reference.* See [GCL92] theorem 8.1. □

From this theorem one can obtain a method to count the number of distinct roots of $f$ evaluated at a fixed point.

**Corollary 3.0.6** ([Col75], Theorem 3)**.** *Let $f \in \mathbb{Z}[x_1,\ldots,x_\tau]$, with $x_i$ of domain $\mathbb{R}, 1 \leq i \leq \tau, \alpha := (\alpha_1,\ldots,\alpha_{\tau-1}) \in \mathbb{R}^{\tau-1}$. The number of distinct real roots $c_r$ of $f(\alpha,x_\tau) = f(\alpha_1,\ldots,\alpha_{\tau-1},x_\tau)$ is given by*

$$c_r = deg_{x_\tau}(f(\alpha,x_\tau)) - deg_{x_\tau}(gcd(f(\alpha,x_\tau),f'((\alpha,x_\tau)))).$$

Computing the GCD of polynomials however is not a trivial task, which is where polynomial remainder sequences come into play.

**Definition 3.0.7.** *Let $R$ be a UFD. Given polynomials $f,g \in R[x]$ with $deg(f) = n, deg(g) = m$ $f = \sum_{i=0}^{n} a_i x^i, g = \sum_{i=0}^{m} b_i x^i$ the Sylvester matrix of $f$ and $g$ (with respect to the variable $x$) is defined as*

$$Syl_x(f,g) := \begin{pmatrix} a_n & & \cdots & & a_0 & & & \\ & a_n & & \cdots & & a_0 & & \\ & & \ddots & & & & \ddots & \\ & & & a_n & & \cdots & & a_0 \\ b_m & & \cdots & & b_0 & & & \\ & b_m & & \cdots & & b_0 & & \\ & & \ddots & & & & \ddots & \\ & & & b_m & & \cdots & & b_0 \end{pmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}a\\a\\a\\a\end{matrix}}\right\} m \\ \left.\vphantom{\begin{matrix}a\\a\\a\\a\end{matrix}}\right\} n \end{matrix}$$

**Definition 3.0.8.** *Let $M_{i,j}$ be the the matrix obtained from $Syl_x(f,g)$ by deleting the last $j$ rows of $f$ coefficients, the last $j$ rows of $g$ coefficients and the last $2j+1$ columns except the column $m+n-i-j$.*

1. *The $j$-th subresultant of $f$ and $g$ is defined as $S_j(f,g) := \sum_{i=0}^{j} det(M_{j,i})x^i$.*
2. *The $j$-th principal subresultant coefficient of $f$ and $g$ is the leading coefficient of $S_j(f,g)$, $psc_j(f,g) := det(M_{j,j})$.*
3. *the resultant of $f$ and $g$ is defined as: $res(f,g) := det(Syl_x(f,g)) = S_0 = psc_0(f,g)$.*

The (inductive) euclidean algorithm to compute the GCD of two univariate polynomials $f,g \in \mathbb{Z}[x], deg(f) = n, deg(g) = m, 0 < m \leq n$ relies on the fact that there always exists a remainder polynomial $r \in \mathbb{Z}[x]$ along with $a,b,c \in \mathbb{Z}$ such that $af = bg+cr$ (since $\mathbb{Z}[x]$ is no euclidean ring, calculation is done in $\mathbb{Q}[x]$ and normalized back to $\mathbb{Z}[x]$). One should note that this remainders are only unique up to multiplication with elements of $\mathbb{Z}$. During the euclidean algorithm, subsequent remainders are calculated, starting with $rem(f,g)$, then $rem(g,rem(f,g))$ and so on until the remainder is element of $\mathbb{Z}$. This leads to a sequence of $f_1 = f, f_2 = g, f_3 = rem(f,g), \ldots, f_k = 0$. The GCD of $f$ and $g$ then is $GCD(cont(f),cont(g)) \cdot f_{k-1}$. The problem of calculating the GCD of two polynomials with the standard euclidean algorithm is, that even in the univariate case the coefficients can grow at a rapid rate when using the euclidean algorithm.
Collins discovered in [Col67], that subresultants also define polynomial remainder sequences. Thus subresultants could be used to compute the GCD. The essential information that is of importance here can be deduced from his fundamental theorem of polynomial remainder sequences and formulated as followed.

**Corollary 3.0.9.** *Let $f,g \in \mathbb{Z}[x]$ with $deg(f) \geq deg(g)$. The sequence $f_1,\ldots,f_k = 0$ with $f_1 := f, f_2 := g, f_i := S_{deg(f_{i-1})-1}(f,g), \forall 3 \leq i \leq k$ is a polynomial reminder sequence.*

$$GCD(f,g) = GCD(cont(f),cont(g)) \cdot prim(f_{k-1}).$$
$$S_j(f,g) = 0 \ \forall j \neq deg(f_i) - 1, i \in \{0,\ldots,k-1\}$$

Another concept that is related to subresultant, is the discriminant.

**Definition 3.0.10.** *Let $R$ be a UFD, $f \in R[x]$, let $\alpha_1,\ldots,\alpha_m$ be the zeros of $f$ in the algebraic closure of $R$, $a := ldcf(f)$. The discriminant of $f$ is defined as followed:*

$$discr_x(f) := a^{2m-2} \prod_{i<j} (\alpha_i - \alpha_j)^2$$

The following well known theorem shows the relation to resultants

**Theorem 3.0.11.** *Let $R$ be an integral domain, $f \in R[X]$ $m := deg(f)$, $a := ldcf(f)$ $c := char(R)$. Assume $c$ does not divide $m$, then*

$$a \cdot discr(f) = (-1)^{m(m-1)/2} res(f,f')$$

Another connection between resultants and discriminants can be observed in the product rule for discriminants.

**Theorem 3.0.12.** *Let $R$ be an integral domain with $char(R) = 0$, let $f,g \in R[X]$.*

$$discr(f \cdot g) = discr(f) discr(g) res(f,g)^2$$

*Proof reference.* See theorem 2.3.3 in [McC85]. □

Both resultants and discriminants have a very important property:

**Corollary 3.0.13.** *Let $R$ be a UFD, let $f,g \in R[x]$.*

- *$discr(f) = 0 \Leftrightarrow f$ not square-free.*
- *$res(f,g) = 0 \Leftrightarrow f$ and $g$ not relatively prime.*

# Chapter 4

# The Operators

## 4.1 General Concept

The desired property for a projection operator *proj* is stated as: Any CAD of $\mathbb{R}^{\tau-1}$, that is sign-invariant on $proj(A_\tau)$ induces a CAD of $\mathbb{R}^\tau$, that is sign-invariant on $A_\tau$. Then the inductive projection on to the $\mathbb{R}^1$ can be used to create a CAD of $\mathbb{R}^1$, which recursively induces a CAD of $\mathbb{R}^\tau$.
So according the definition of CAD it suffices for any on $proj(A)$ sign-invariant region to induce an algebraic stack, that is sign-invariant on $A_\tau$. An intuitive way of building such a stack given a region $R$ is to look at the real variety $V(f) := \{(a_1,\ldots,a_\tau) \in \mathbb{R}^\tau \mid f(a_1,\ldots,a_\tau) = 0\}$ of each polynomial $f \in A_\tau$. If $Z(R) \cap \bigcup_{f\in A_\tau} V(f)$ forms a stack over $R$, then every sample point of $R$ can be lifted to a set of sample point of $Z(R)$, representing the sign-invariant regions of $A_\tau$ on $Z(R)$. This regions are the sectors and sections of the functions, that define the stack on $Z(R) \cap \bigcup_{f\in A_\tau} V(f)$.
However to ensure this, for any function $f \in A_\tau$, $V(f)$ needs to only consist of $k$ disjoint sections on $Z(R)$ for some $k \geq 0$. A function with that property is called delineable. Note that while Collins proposed a more strict definition of delineability in [Col75], he argued that this relaxed definition is sufficient in [ACM84]. If additionally those sections of two different functions $f,g \in A, f \neq g$ are either disjoint or identical, then it gives rise to a stack over $R$ determined by the continuous functions whose graphs describe $Z(R) \cap \bigcup_{f\in A} V(f)$.
Figure 4.1 illustrates some problematic scenarios that could occur. It shows some region $R$ and the portion of real variety of 3 hypothetical bivariate polynomials on this region. While the variety of $f_1$ on $R$ can be described by 2 disjoint continuous functions and therefore is delineable on $R$, one of the continuous function graphs crosses the graph of the continuous function that describes $V(f_2)$. Thus no sign-invariant stack over $R$ is induced with respect to $\{f_1,f_2\}$. The variety of $f_3$ on $R$ can not be described by disjoint continuous functions as it contains a self-crossing, therefore $f_3$ is not delineable and no on $f_3$ sign-invariant stack on $R$ can be constructed. The right side of figure 4.1 shows, how $R$ needs to be split up in order to obtain regions, where delineability and the disjoint section criteria is preserved on the region $R_1,\ldots,R_5$.

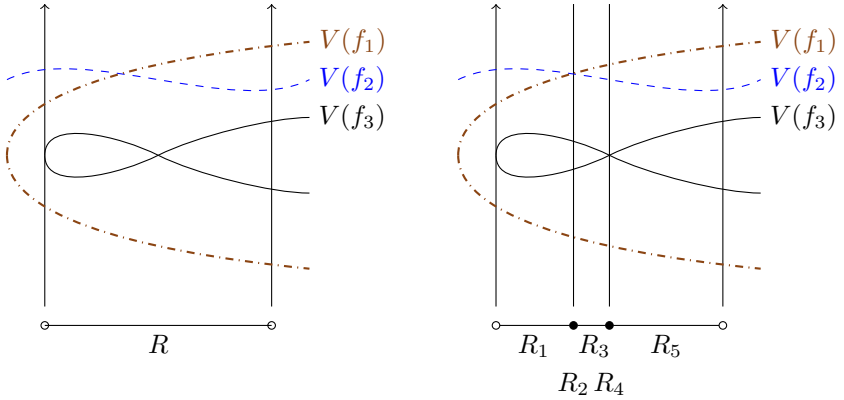


Figure 4.1: Left: Only $V(f_1)$ and $V(f_2)$ delineable on $R$. Right: $V(f_1)$,$V(f_2)$ and $V(f_3)$ delineable on $R_1$-$R_5$.

Self-crossing is only one example case where the variety portion on a region is not delineable, there are many more, like cusps or asymptotes, where continuous functions can not describe the variety. One special case however is, when a function vanishes on every point of a region. A simple example is, when looking at the set $A_2 := \{f(x,y) := x, g(x,y) := x^2 + y^2 - 1\} \subset \mathbb{Z}[x,y]$. $f(0,y) = 0, \forall y \in \mathbb{R}$ so $f$ is not delineable on the region $R = \{0\}$ nor on any superset of it. This is no problem since $f$ is sign-invariant on $Z(R)$ and therefore can be ignored when decomposing $Z(R)$. $f$ is called identically zero on $R$.
So, given a finite set of polynomials $A_\tau$, the idea behind the *proj* operator is to make sure the following 2 conditions hold for any $proj(A_\tau)$ invariant region $R$:

1. $\forall f \in A_\tau : f$ is delineable or identically zero on $R$.
2. $\forall f,g \in A_\tau, f \neq g$ the sections of $V(f)$ and $V(g)$ on $Z(R)$ are either disjoint or identical.

The projection operators that are compared in this paper, therefore consist of 2 parts each, one to satisfy each condition.

## 4.2 Definition of the Operators

### 4.2.1 Collins

Given a set of polynomials $A_\tau \subset \mathbb{Z}[x_1, \ldots, x_\tau]$ as input with main variable $x_\tau$, a projection operator produces a set of polynomials $A_{\tau-1} \subset \mathbb{Z}[x_1, \ldots, x_{\tau-1}]$ as output with main variable $x_{\tau-1}$. Collins defined his operator using his definition of principle subresultant coefficient sets and reducta sets.

**Definition 4.2.1.** *Let $f,g \in A_\tau$ with $n := min(deg(f), deg(g))$. The psc set of $f$ and $g$ is defined as:*

$$PSC(f,g) := \{psc_j(f,g) \mid 0 \leq j \leq n, psc_j(f,g) \neq 0\}$$

.

**Definition 4.2.2.** *Let $f \in A_\tau$ with $deg(f) = n$. The reducta set of $f$ is defined as:*

$$RED(f) := \{red^k(f) \mid 0 \leq k \leq n, red^k(f) \neq 0\},$$

*where* $red^0_{x_\tau}(f) := f, red^k_{x_\tau}(f) := red_{x_\tau}(red^{k-1}_{x_\tau}(f))$.

His operator as defined in [ACM84] is:

**Definition 4.2.3** (Collins)**.** *Let $A_\tau$ be a finite set of $\tau$-variate polynomials with $A_\tau = \{f_1, \ldots, f_m\}$.*

$$\begin{aligned} projC(A_\tau) &:= projC_1(A_\tau) \cup projC_2(A_\tau) \\ projC_1(A_\tau) &:= \bigcup_{f_i \in A_\tau} \bigcup_{r \in RED(f_i)} (\{ldcf(r)\} \cup PSC(r,r')) \\ projC_2(A_\tau) &:= \bigcup_{1 \leq i < j \leq m} \bigcup_{\substack{r_i \in RED(f_i) \\ r_j \in RED(f_j)}} PSC(r_i,r_j) \end{aligned}$$

He proved, that given any on $projC_1(A_\tau)$ sign-invariant region $R$, the polynomials in $A_\tau$ are all either delineable or identically zero on $R$. In addition he then showed that if $projC_1$ gets extended with $projC_2$ to $projC$, that the sections of $V(f)$ and $V(g)$ of any two polynomials $f,g \in A_\tau$ are either disjoint or identical on any on $projC(A_\tau)$ sign-invariant region $R$. Then he stated out a method of obtaining defining formulas on $Z(R)$ given a semi-algebraic region $R$, that is sign-invariant on $projC$. This concludes the proof that indeed an algebraic decomposition can be established within the CAD procedure, because in the base phase, the regions are connected subsets of $\mathbb{R}$ and thus easily definable. In [Col75] Collins also analyzed his operator with the result, that, given $|A_\tau| = m$ and a bound on the degree of each polynomial in its main variable of $n$, the number of polynomials in $projC(A_\tau)$ is dominated by $m^2n^3$.

### 4.2.2 Hong

Hong showed in [Hon90], that within $ProjC_2$ the added *psc* sets have redundancies and proposed the following operator:

**Definition 4.2.4** (Hong)**.** *Let $A_\tau$ be a finite set of $\tau$-variate polynomials with $A_\tau = \{f_1, \ldots, f_m\}$.*

$$\begin{aligned} projH(A_\tau) &:= projC_1(A_\tau) \cup projH_2(A_\tau) \\ projH_2(A_\tau) &:= \bigcup_{1 \leq i < j \leq m} \bigcup_{r \in RED(f_i)} PSC(r,f_j) \end{aligned}$$

Again, let $|A_\tau| = m$ and $n$ be a bound on the degree of each polynomial in its main variable, Hong verified, that the number of polynomials included in $projH(A_\tau)$ is dominated by $m^2n^2$, which implies a possible significant impact on the overall performance compared to Collins operator, since the degree bound impacts the size in quadratic instead of cubic order.

### 4.2.3 McCallum

While Hongs operator can be validated with the same arguments that Collins used, McCallum defined an operator in [McC85] based on the work of Zariski on local properties of complex hyper-surfaces in [Zar75]. McCallum found this work to be closely related to the properties that projection operators in the CAD procedure have. His operator ensures analytic delineability, which is defined as the usual delineability property but instead of continuous functions to describe the real variety, analytic functions are used (which are special continuous functions). For exact definitions of analytic functions and their properties see chapter 2.1 of [McC85]. Also his operator aims at a decomposition into connected submanifolds (subsets of the $\mathbb{R}^\tau$ with special properties) instead of regions. For the definition of submanifolds the reader is referred to chapter 2.2 of [McC85], since the knowledge in algebraic geometry is not necessary for this thesis. The CAD procedure stays untouched regardless of these theoretical changes in McCallums operator. A relevant change in McCallums work however is, that order-invariance is guaranteed when creating a CAD with his projection operator, which is a stronger proerty than sign-invariance.

**Definition 4.2.5.** *$f \in \mathbb{Z}[x_1,\ldots,x_\tau], \alpha = (\alpha_1,\ldots,\alpha_\tau) \in \mathbb{R}^\tau$. $f$ has order $k$ in $\alpha$, written $ord_\alpha(f) = k$, if $k \in \mathbb{N}$ is minimal such that some $k$-th derivative $\frac{\partial^k f}{\partial x_1^{i_1},\ldots,\partial x_\tau^{i_\tau}}$, $i_1,\ldots,i_\tau \in \mathbb{N}, \sum_{j=1}^{\tau} i_j = k$ of $f$ does not vanish at $\alpha$.*
*Let $S \subset \mathbb{R}^\tau$ be a connected submanifold, $f$ is called order-invariant on $S$, if for a $k \in \mathbb{N}$ $ord_\alpha(f) = k, \forall \alpha \in S$.*

He defined his operator as:

**Definition 4.2.6** (McCallum)**.** *Let $A_\tau$ be a finite set of $\tau$-variate polynomials and let $B_\tau$ be the finest square-free base of $A_\tau$ with $B_\tau = \{f_1,\ldots,f_m\}$.*

$$projM(A_\tau) := projM_1(B_\tau) \cup projM_2(B_\tau) \cup \bigcup_{f_i \in A_\tau} \{cont_{x_\tau}(f_i)\}$$

$$projM_1(B_\tau) := \bigcup_{f_i \in B_\tau} \left( \{discr(f_i)\} \cup \bigcup_{r \in RED(f_i)} \{ldcf(r)\} \right)$$

$$projM_2(B_\tau) := \bigcup_{1 \le i < j \le m} \{res(f_i, f_j)\}$$

Although it seems substantially different to the previous operators, McCallum actually showed that, when using 3.0.11, one could also redefine Collins operator to operate with subdiscriminants rather than subresultants such hat $projM \subseteq projC$. For more details see chapter 3.1 in [McC85]. McCallums operator not only requires the use of finest square-free bases in each step, but also can only guarantee the correctness of a CAD procedure with his operator when the input polynomials are well-oriented.

**Definition 4.2.7.** *A finite set $A_\tau$ of non-zero $\tau$-variate integral polynomials is called **well-oriented** with regards to an projection operator $P$, if $\tau = 1$ or, if $\tau > 1$, then*

1. *for every $f \in prim(A_\tau)$, $f(\alpha, x_\tau) = 0$ for at most finitely many $\alpha \in \mathbb{R}^{\tau-1}$ and*
2. *$P(A_\tau)$ is well-oriented.*

In addition to this restriction, the finite amount of real roots of the well oriented polynomials in the set $projM(A)$ require a slight modification of the lifting phase to preserve the order-invariance in every step. McCallum also provided incomplete methods to check, whether a polynomial has finitely many zeros or not. It is also possible to detect if the input polynomials were not well-oriented during lifting phase, when one keeps track while creating sample points, whether they present a space of positive dimension or not. If a sample point $p \in \mathbb{R}^i$, that represents a cell of positive dimension (thus was chosen to represent a sector in some lower dimensional lifting step) causes a $(i+1)$-variate projection polynomial to vanish, then this polynomial has an infinite amount of zeros (namely all sample points that could have been chosen instead of the one that lead to $p$ in this lower dimensional lifting step). McCallum further argued, that the vast majority of polynomials is well-oriented including all polynomials with at most three variables. A test on a set of 7317 Benchmark from the SMT-COMP sowed, that in 353 cases, some projection set turned out to not be well-oriented. This however causes no failure in the actual solving process on the examples, when using SMT-RAT, because those benchmarks were either unsatisfiable by their boolean structure or even satisfiable by the broken CAD that McCallums operator produces.

### 4.2.4 Brown

While McCallums operator produces order-invariant CADs which might be needed in some scenarios, it is not required to solve QF_NRA formulas, because only the sign-invariance is mandatory there. Brown observed, that then McCallums operator can be further improved. He showed, that it is sufficient for the $projM_1$ operator to only contain the leading coefficient of each polynomial instead of all coefficients. This however needs to come along with further modifications in the liftings stage, that are explained in chapter 5.4.

**Definition 4.2.8** (Brown)**.** *Let $A_\tau$ be a finite set of $\tau$-variate polynomials and let $B_\tau$ be the finest square-free base of $A$ with $B_\tau = \{f_1, \ldots, f_m\}$.*

$$projB(A_\tau) := projB_1(B_\tau) \cup projM_2(B_\tau) \cup \bigcup_{f_i \in A_\tau} \{cont(f_i)\}$$

$$projB_1(B_\tau) := \bigcup_{f_i \in A_\tau} \{discr(f_i)\} \cup \{ldcf(f_i)\}$$

He also made another modification to the lifting stage, that allows the CAD to stay correct in some cases, where the polynomials are not well-oriented.
Note that essentially $projB \subseteq projM \subseteq projH \subseteq projC$.

# Chapter 5

# On the Theory behind the Operators

## 5.1 Proof overview for Collins operator

Recall the desired properties of projection operators given in chapter 4.1. Collins proved that $projC_1$ ensures, given a finite set of polynomials $A_\tau \subset \mathbb{Z}[x_1,\ldots,x_\tau]$ and region $R$, where on $projC_1(A_\tau)$ is sign-invariant on, that every $f \in A_\tau$ is delineable on $R$. Core of his argumentation was the following theorem, that states the connection between the (point-wise) number of distinct roots of a polynomial and its delineability.

**Theorem 5.1.1.** *Let $f \in \mathbb{Z}[x_1,\ldots,x_\tau]$ be a polynomial, let $\tau \geq 2$. Let $R$ be a connected subset of $\mathbb{R}^{\tau-1}$. If $ldcf(f(\alpha_1,\ldots,\alpha_{\tau-1},x_\tau)) \neq 0, \forall \alpha = (\alpha_1,\ldots,\alpha_{\tau-1}) \in R$ and it exists an $l \in \mathbb{N}$, such that the number of distinct roots of $f(\alpha_1,\ldots,\alpha_{\tau-1},x_\tau)$ is $l, \forall \alpha = (\alpha_1,\ldots,\alpha_{\tau-1}) \in R$, then $f$ is delineable on $R$.*

*Proof reference.* See [Col75] theorem 1 (Note the more strict definition of delineability). □

It could happen, that certain coefficients vanish on a set of points, like this example shows:

**Example 5.1.2.** *Given $f(x,y,z) = (x^2+y^2-1)z^2 + (x-1)z + (x-1)^2 + y^2$ with main variable $z$. Consider $P_1 = \{(x,y) \in \mathbb{R}^2 \mid x^2+y^2-1=0\}, P_2 = \mathbb{R}^2 \setminus P_1$.*

- $ldcf(f(\alpha,z)) = 0, \forall \alpha \in P_1$
- $deg(f(\alpha,z)) = 1, \forall \alpha \in P_1 \setminus \{(1,0)\}$
- $f(1,0,z) = 0$
- $deg(f(\alpha,z)) = 2, \forall \alpha \in P_2$

To ensure that $f$ has a constant degree on any on $proj(f)$ sign-invariant region it suffices to add all coefficients of $f$ to the projection. It becomes a bit more complicated to keep the number of distinct roots constant and this is ensured the following way:

Assume a region $R \subset \mathbb{R}^{\tau-1}$ where all coefficients of $f$ are sign-invariant on. Further let $k$ denote the integer, such that the first $k$ coefficients of $f$ vanish on $R$, $0 \leq k \leq deg(f)$. Then $f(\alpha,x_\tau) = g(\alpha,x_\tau),\forall \alpha \in R$, where $g := red^k(f)$. So it suffices to ensure the delineability of $g$ on $R$. If $g$ is identically zero on $R$, $g$ can be ignored, else $ldcf(g(\alpha,x_\tau)) \neq 0,\forall \alpha \in R$. Thus the requirements of theorem 5.1.1 are almost met, the number of distinct roots of $g$ could still vary on $R$.
As corollary 3.0.6 states, the number of distinct roots $c_r$ in a point $\alpha \in R$ of $g$ is given by $c_r = deg(g(\alpha,x_\tau)) - deg(GCD(g(\alpha,x_\tau),g'(\alpha,x_\tau)))$. Corollary 3.0.9 implies that, given a point $\alpha \in R$, $GCD(g(\alpha,x_\tau),g'(\alpha,x_\tau))$ is the least $j$-th subresultant $S_j(g(\alpha,x_\tau),g'(\alpha,x_\tau))$, which is not identically zero. Since $ldcf(g) \neq 0$ and $ldcf(g') \neq 0$ the Sylvester matrix $Syl(g(\alpha,x_\tau),g'(\alpha,x_\tau)) = Syl(g,g')(\alpha,x_\tau)$, so per definition of subresultants it does not matter, if the j-th subresultant is calculated on $g$ and $g'$ and then get evaluated at $\alpha$ afterwards, or if the subresultant of $g(\alpha,x_\tau)$ is constructed, thus: $S_j(g,g')(\alpha,x_\tau) = S_j(g(\alpha,x_\tau),g'(\alpha,x_\tau))(x_\tau)$. It concludes, that the same property holds for the principle subresultant coefficients: $psc_j(g,g')(\alpha) = psc_j(g(\alpha,x_\tau),g'(\alpha,x_\tau))$.
So it would suffice to ensure that all subresultants of $g$ and $g'$ are of constant degree on $R$. It is even sufficient to guarantee that all $psc_j(g,g')$ are of constant degree on $R$, as the following corollary shows.

**Corollary 5.1.3.** *Let $f,g \in \mathbb{Z}[x_1,\ldots,x_\tau]$, then $deg(GCD(f,g)) = k$ if and only if $k$ is the least $j$ such that $psc_j(f,g) \neq 0$.*

*Proof reference.* See [ACM82] See corollary 3.3 and theorem 3.2. □

This is why in addition to all coefficients the *psc*s for every possible reductum of $f$ are added to $projC_1$, which then is validated to provide delineability for polynomials $f$ on each on $projC_1(f)$ sign-invariant region.
To further guarantee that any on $projC(A_\tau)$ sign-invariant region $R$ the sections (the delineating functions) of different $f \neq g, f,g \in A_\tau$ do not cross, Collins argued, that it would suffice to make sure, that the product $P_{A_\tau} := \prod_{f_i \in A_\tau} f$ is delineable on $R$. This is due to the following observation. Assume $P_{A_\tau}$ as well as all polynomials in $A_\tau$ to be delineable on $R$. Let $\alpha = (\alpha_1,\ldots,\alpha_{\tau-1}) \in R$ and let $\beta \in \mathbb{R}$ with $P_A(\alpha_1,\ldots,\alpha_{\tau-1},\beta) = 0$. Since $P_{A_\tau} \in \mathbb{Z}[x_1,\ldots,x_\tau]$ and $\mathbb{Z}[x_1,\ldots,x_\tau]$ is a UFD, $P_A(\alpha_1,\ldots,\alpha_{\tau-1},\beta) = 0$ if and only if there exists $f_i \in A_\tau$ with $f_i(\alpha_1,\ldots,\alpha_{\tau-1},\beta) = 0$. Since $f_i$ is delineable on $R$, there exists a delineating function $\theta_{f_i}$ of $f_i$, with $\theta_{f_i}(\alpha_1,\ldots,\alpha_{\tau-1}) = \beta$. On the other hand assume any point $\gamma \in Im(\theta_{f_i})$. Then there exists $\alpha' = (\alpha'_1,\ldots,\alpha'_{\tau-1}) \in R$ with $P_{A_\tau}(\alpha'_1,\ldots,\alpha'_{\tau-1},\gamma) = 0$. It concludes that $\theta_{f_i}$ is a delineating function of $P_{A_\tau}$. Since $P_{A_\tau}$ is delineable on $R$, the delineating functions $\theta_{P_{A_\tau}}$ are disjoint, thus the delineating functions of $f_i$ and $f_j$ are either disjoint or identical. The delineability of the functions in $A_\tau$ is already covered by $projC_1$.

The only problematic scenario remaining is, when delineating functions cross each other. Let $f,g \in A_\tau$ and let $R$ be a region, where $projC_1(A_\tau)$ is sign-invariant on. Assume that a delineating function of $f$ crosses a delineating function of $g$. Let $(\alpha_1,\ldots,\alpha_{\tau-1},\beta) \in Z(R)$ with $\alpha = (\alpha_1,\ldots,\alpha_{\tau-1}) \in R$ and $\beta \in \mathbb{R}$ be the point in $Z(R)$, where those delineating functions are equal. Then $f(\alpha_1,\ldots,\alpha_{\tau-1},\beta) = 0$ and $g(\alpha_1,\ldots,\alpha_{\tau-1},\beta) = 0$, thus $(x_\tau - \beta)$ divides $f(\alpha,x_\tau)$ and $(x_\tau - \beta)$ divides $g(\alpha,x_\tau)$, which implies that $(x_\tau - \beta)$ divides $GCD(f(\alpha,x_\tau),g(\alpha,x_\tau))$.
Conversely consider $\gamma \in \mathbb{R}$. If $(x_\tau - \gamma)$ devides $GCD(f(\alpha,x_\tau),g(\alpha,x_\tau))$, then $f(\alpha,\gamma) =$

0 and $g(\alpha,\gamma) = 0$ so some delineating functions of $f$ and $g$ cross each other. Example 5.1 depicts this situation.

**Example 5.1.4.** *Let* $f(x,y) := y - x$ *and* $g(x,y) := y - 3$*, with main variable* $y$*. Let* $\alpha_1 := 1, \alpha_2 := 3, \beta := 3, \gamma_1 := 1, \gamma_2 := 3$*. Then*

$$GCD(f(\alpha_1,y),g(\alpha_1,y)) = GCD(y-1,y-3) = 1,$$
$$GCD(f(\alpha_2,y),g(\alpha_2,y)) = GCD(y-3,y-3) = y-3.$$

*Figure 5.1 illustrates the delineating functions of* $f$ *and* $g$*.*

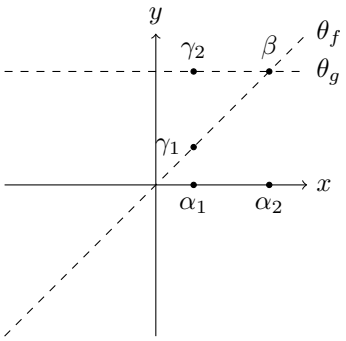


Figure 5.1: $\{f,g\}$-invariant CAD of $\mathbb{R}$

Collins therefore argued, that if the degree of the GCD of pairs of polynomials $f$ and $g$ is constant on each $projC$ sign-invariant region, then the delineating functions of $f$ and $g$ are either disjoint or identical on those regions. Using corollary 5.1.3 and the same argumentation as for the validation of $projC_1$ this concludes to taking pairs of functions out of the reducta sets of $f,g \in A_\tau$ and adding their *psc* sets to the projection. Then for some reducta $r_i \in RED(f), r_j \in RED(g)$ $r_i = f$ and $r_j = g$ on any on $projC(A_\tau)$ sign-invariant region $R$. So $ldcf(r_i) \neq 0$ and $ldcf(r_j) \neq 0$ and $deg(psc_k(r_i,r_j)) = deg(GCD(r_i,r_j))$ for some $0 \leq k \leq min(deg(r_i),deg(r_j)))$. Thus the degree of the $GCD$ is constant on $R$ and therefore the delineating functions of $r_i$ and $r_j$ are either disjoint or identical on $R$. The full proof is given in [ACM82] theorem 3.7.

## 5.2 Correctness of Hongs Refinement

While Collins used the observation that given $f,g \in \mathbb{Z}[x_1,\ldots,x_\tau]$ and $\alpha \in \mathbb{R}^{\tau-1}$, $Syl(f,g)(\alpha,x_\tau) = Syl(f(\alpha,x_\tau),g(\alpha,x_\tau))$ only when $ldcf(f)$ and $ldcf(g)$ do not vanish in $\alpha$, Hong came up with the following lemma in [Hon90] that can be deduced by the definition of *psc*s.

**Lemma 5.2.1.** *Let* $f,g \in \mathbb{Z}[x_1,\ldots,x_\tau], deg_\tau(f) = n, deg_\tau(g) = m$ *and let* $\alpha \in \mathbb{R}^{\tau-1}$ *such that* $ldcf(f(\alpha)) \neq 0$ *and* $deg(g(\alpha)) = l \geq 1$ *then*

$$psc_j(f,g)(\alpha) = [ldcf(f)(\alpha)]^{m-l}\, psc_j(f(\alpha,x_\tau),g(\alpha,x_\tau))$$

*Proof.* $psc_j(f(\alpha,x_\tau),g(\alpha,x_\tau))$ is defined to be the determinant of the Minor $M_{j,j}$ of $Syl(f(\alpha,x_\tau),g(\alpha,x_\tau))$.

$$M_{j,j} := \left(\begin{array}{cccccc} a_n & & \cdots & & a_0 & \\ & a_n & & \cdots & & a_0 \\ & & \ddots & & & & \ddots \\ & & & a_n & & \cdots & a_j \\ b_l & & \cdots & & b_0 & \\ & b_l & & \cdots & & b_0 \\ & & \ddots & & & & \ddots \\ & & & b_l & & \cdots & b_j \end{array}\right) \begin{array}{l} \left.\vphantom{\begin{array}{c}a\\a\\a\\a\\a\\a\\a\end{array}}\right\} \text{l-j} \\ \left.\vphantom{\begin{array}{c}a\\a\\a\\a\\a\\a\\a\end{array}}\right\} \text{n-j} \end{array}$$

Now assume $deg(g(\alpha,x_\tau)) = l$, let $M'_{j,j}$ be the $j$-th minor of $Syl(f,g)(\alpha,x_\tau)$.

$$M'_{j,j} := \left(\begin{array}{ccc:ccc} a_n & \cdots & & a_0 & & \\ & \ddots & & & \ddots & \\ & & a_n & & & a_0 \\ \hdashline 0 & \cdots & 0 & & & \\ 0 & \cdots & 0 & & M_{j,j} & \\ 0 & \cdots & 0 & & & \end{array}\right) \begin{array}{l} \left.\vphantom{\begin{array}{c}a\\a\\a\\a\\a\\a\\a\end{array}}\right\} \text{m-l} \\ \left.\vphantom{\begin{array}{c}a\\a\\a\\a\\a\\a\\a\end{array}}\right\} \text{n+l-2j} \end{array}$$

By inductively using the well known Laplace expansion along the first column to calculate $det(M'_{j,j})$ the claim follows. □

The consequence of this lemma is, that Collins operator can be reduced without losing functionality.

Given a finite polynomial set $A_\tau \in \mathbb{Z}[x_1,\ldots,x_\tau]$ let $f_i,f_j \in A_\tau$ with $deg(f_i) = n$, $deg(f_j) = m$, $projC_2$ constructs every possible combination of reducta pairs of $f_i$ and $f_j$ and adds their $psc$ sets to ensure they are all constant on any on $projC(A_\tau)$ sign-invariant region $R$. Then there are reducta $r_k \in RED(f_i)$ and $r_l \in RED(f_j)$ with $r_k(\alpha,x_\tau) = f_i(\alpha,x_\tau)$ and $r_l(\alpha,x_\tau) = f_j(\alpha,x_\tau)$ and with $ldcf(r_k(\alpha,x_\tau)) \neq 0$ and $ldcf(r_l(\alpha,x_\tau)) \neq 0$ for all $\alpha \in R$. With the lemma it is clear, that this can be optimized. It suffices to construct the reducta of $f_i$, as it is done in $projH$. Then on any on $projH(A_\tau)$ sign-invariant region $R$, all elements of $PSC(r_i,f_j)$ are constant on $R$, thus $[ldcf(r_i)(\alpha)]^{m-l}\, psc_k(r_i(\alpha,x_\tau),f_j(\alpha,x_\tau))$ is constant for $0 \leq k \leq min(n,l)$ where $l = deg(r_i)$, since $ldcf(r_i)$ is non-negative and constant (due to $projC_1 \subset projH$) on $R$. Thus also $psc_k(r_i(\alpha,x_\tau),f_j(\alpha,x_\tau))$ is constant. This was essential for the functionality of $projC$, so $projH$ works as intended, while producing less polynomials in each projection step.

## 5.3 Changes, when using McCallums Operator

McCallums work in [McC85] centralizes on his main theorem that he stated as:

**Theorem 5.3.1** (Lifting Theorem)**.** *Let $f \in \mathbb{Z}[x_1,\ldots,x_\tau]$ with $deg_{x_\tau} \geq 1$ and with $discr_{x_\tau}(f) \neq 0$. Let $S$ be a connected submanifold of $\mathbb{R}^{\tau-1}$ in which $f$ has constant degree and is not identically zero, and in which $discr_{x_\tau}(f)$ is order-invariant. Then $f$ is analytic delineable on $S$ and is order-invariant in each of its sections over $S$.*

*Proof reference.* See theorem 3.2.1, proof contained in chapter 3.3 of [McC85]. □

Note that the sectors between two sections of the variety of $f$ on $S$ are also order-invariant since the order is 0.

The lifting theorem can be used to validate his operator. Let $A_\tau \subset \mathbb{Z}[x_1,\ldots,x_\tau]$ be the finest square-free base of a finite set of polynomials, let $A_\tau = \{f_1,\ldots,f_n\}$. The lifting theorem already ensures the analytic delineability of the polynomials on any on $projM_1$ order-invariant submanifold, because $projM_1$ consists of all coefficients (to keep the degree invariant) and the discriminants of all polynomials of $A_\tau$. Since $A_\tau$ is per assumption a finest square-free base, $discr(f) \neq 0, \forall f \in A_\tau$ as corollary 3.0.13 implies. The only exception is, when a polynomial or its discriminant is identically zero on $S$. Assuming well-oriented polynomials as input, this would mean that $S$ is zero dimensional. Then the lifting theorem can not be applied. However if $S$ has dimension zero, the polynomials are either delineable or identically zero on $S$. The order-invariance of a function $f_i \in A_\tau$ is not guaranteed on the cylinder over $S$, when $f_i$ is identically zero. Example 5.3.2 depicts such a scenario. The solution for this problem is, to ensure order-invariance in the lifting phase, when some polynomial vanishes on a sample point. More details on how to establish order-invariance in this special case are given later.

To also guarantee that delineating functions of different polynomials in $A_\tau$ do not cross on a on $projM(A_\tau)$ order-invariant submanifold $S$, McCallum argued like Collins and proved by using the lifting theorem on $P_{A_\tau} := \prod_{f \in A_\tau} f$, that $P_{A_\tau}$ is also analytic delineable. Using theorem 3.0.12:

$$discr(P_{A_\tau}) = \prod_{i=1}^{n} discr(f_i) \prod_{j=i+1}^{n} res(f_i, f_j)^2.$$

Since pairwise all resultants are included in $projM$, and also the discriminants of each polynomial are in $projM$, all of the factors are order-invariant on $S$. According to [McC85] lemma 3.2.2 this implies that also the product, thus $discr(P_{A_\tau})$ is order invariant on $S$. The requirements of the lifting theorem are therefore met, with exception to the side-case, where $P_{A_\tau}$ vanishes everywhere on $S$ or its discriminant is zero. The former is no problem, because [McC85] lemma 3.2.2 works in both directions and thus if all polynomials in $A_\tau$ are order-invariant on $S$, also $P_{A_\tau}$ is order-invariant on $S$. This is given, as the lifting phase will deal with those exceptional cases. The latter is also no problem as of corollary 3.0.13 implies. Since only finest square-free bases are used for projection, the discriminants of the projection polynomial is never 0 as well as pairwise resultants. So the only thing missing is a way to reestablish order-invariance, when some polynomial vanishes identically over a zero dimensional submanifold.
The following example depicts a situation, where order-invariance needs to be established in the lifting phase.

**Example 5.3.2.** *Let $f := (w^2 + x^2 - 1)y + (w - 1), p = f + z^2$. $A_4 := \{p\}$ with variable order $w < x < y < z$. $A_4$ is square-free and since $|A_4| = 1$, it is also a finest square-free base. Note that polynomials in this example all get normalized, thus only the primitive part is considered. Also constants get removed on each projection level and polynomials that only are constant with respect to the main variable are pushed*

*down to the next relevant level.*
*Projection phase:*

- $discr_z(p) = f$
- $ProjM(A_4) = \{f,1\} \rightarrow \{f\} := A_3$. *f is square-free.*
- $discr_y(f) = 1$.
- $ProjM(A_3) = \{g := w^2 + x^2 - 1, h := w - 1, 1\} \rightarrow \{g,\} := A_2$. *h gets pushed down to* $A_1$
- $discr_x(w^2 + x^2 - 1) = w^2 - 1$
- $ProjM(A_2) = \{w^2 - 1, 1, -1\} \rightarrow \{w^2 - 1, h\} := A_1$

*Base phase:*

- $S_1 = (-\infty, -1), S_2 = [-1, -1], S_3 = (-1,1), S_4 = [1,1], S_5 = (1,\infty)$
- $\rightarrow CAD(A_1) = \{-2 - 1, 0, 1, 2\}$

*Lifting phase:*

- $g(0,x) = x^2 - 1 \rightarrow$ *add* $x = 1, x = -1$.
- $g(1,x) = g(-1,x) = x^2 \rightarrow$ *add* $x = 0$.
- $g(2,x) = g(-2,x) = x^2 + 3 \rightarrow$ *add nothing.*
- $CAD(A_2) = \{(0,0),(0,1),(0,2),(0,-1),(0,-2),$
  $(1,0),(1,1),(1,-1),(-1,0),(-1,1),(-1,-1),(2,0),(-2,0)\}$
- $f(1,0,y) = 0 \rightarrow$ *exceptional case.*

*Problem:*

- $f(\alpha,y)$ *is identically zero for some* $\alpha \in CAD(A_2)$. *Consider* $\alpha = (1,0)$.
- *f might not be order-invariant on* $Z((1,0))$.
  - *f is the discriminant of p.*
  - *Lifting theorem can not guarantee, that p is delineable on any submanifold of* $\alpha \times \mathbb{R}$.
- $ord_{(1,0,y)}(f) > 0$, *because f is identically zero on.*
- $ord_{(0,1,y)}(f) \geq 1$, *because:*
  - $d_1 := \dfrac{\partial f}{\partial y} = x - 1 \rightarrow d_1(1,0,y) = 0$
  - $d_2 := \dfrac{\partial f}{\partial x} = 2xy \rightarrow d_2(1,0,y) = 0$
  - $d_3 := \dfrac{\partial f}{\partial w} = 2wy + 1 \rightarrow d_2(1,0,y) = 2y - 1$
- $ord_{(0,1,1)}(f) = 1, ord_{(0,1,0.5)}(f) > 1$

Looking now at the sample point (1,0), $f(1,0,y) = 0$. Order invariance is not given on any submanifold $S \subset (0,1) \times \mathbb{R}$. $f$ is the discriminant of $p$, thus the lifting theorem can not be used to validate, that $p$ is analytic delineable on $S$. The solution to this dilemma is to ensure order-invariance on those submanifolds manually.

Let $f \in A_i$ be a polynomial, that vanishes on a sample point $\alpha \in A_{i-1}$. Let $k$ be the least non-negative integer, such that the set

$$D_k(f) = \left\{ q \mid q \neq 0, \exists e_1, \ldots, e_i. \sum_{r=1}^{i} e_r = k, q = \frac{\partial^k f}{\partial x_1^{e_1}, \ldots, \partial x_i^{e_i}} \right\}$$

is not empty. Then $ord_{(\alpha,\beta)}(f) \geq k, \forall \beta \in \mathbb{R}$ and $ord_{(\alpha,\beta)}(f) > k$ if and only if $D_k(f(\alpha,\beta))$ is empty. Since $f$ is assumed to be well-oriented, there are at most finitely many points $\beta_1, \ldots, \beta_n \in \mathbb{R}$, where $ord_{(\alpha,\beta_i)}(f) > k, 1 \leq i \leq n$. Adding a polynomial with $\beta_1, \ldots, \beta_n$ as zeros to $A_i$ ensures that all those cases are covered by the decomposition, because the points $(\alpha,\beta) \in \mathbb{R}^i$ are chosen as sample points as well as sample points from the open subsets in between. McCallum named this polynomial a delineating polynomial. To determine $\beta_1, \ldots, \beta_n$ McCallum stated, that it would be sufficient to add one $k-th$ partial derivate $d \in D_k(f)$. This would split the cylinder over $\alpha$ according to all of $d$-s zeros. Splitting along all zeros is not necessary, since some other $k-th$ partial derivate might not vanish when $d$ does. Thus Brown suggested in [Bro01b] to instead add a minimal delineating polynomial which is the greatest common divisor of all polynomials in $D_k$. Then the splitting is only done along the points where the order actually changes. This could save a lot of time as the amount of sample points grow exponentially while lifting.

## 5.4 Browns improvement on McCallums Operator

Brown observed in [Bro01a], that the degree-invariance, required in McCallums lifting theorem is already implicitly given by the order-invariance of its discriminant and the sign-invariance of its leading coefficients. This allowed him to state a theorem to justify an improvement in the projection size.

**Theorem 5.4.1.** *Let $f \in \mathbb{Z}[x_1, \ldots, x_\tau]$ with $deg_{x_\tau} \geq 1$ and with $discr_{x_\tau}(f) \neq 0$. Let $S$ be a connected submanifold of $\mathbb{R}^{\tau-1}$ in which $ldcf(f)$ is sign-invariant and where $f$ is not identically zero on any point of $S$, and in which $discr_{x_\tau}(f)$ is order-invariant. Then $f$ has constant degree on $S$.*

*Proof reference.* See theorem 3.1 of [Bro01a]. □

This theorem implies, that it is not necessary to add all coefficients of a polynomial into the projected set. Instead the leading coefficient is sufficient. Exceptions are $\tau$-variate polynomials that turn out to be identically zero on some points of $\mathbb{R}^{\tau-1}$. This sounds problematic at first, but these points, where a polynomial turns out to be identically zero on $S$, can be computed and added manually in the lifting phase. Since input polynomials are assumed to be well-oriented, this are at most finitely many points. Let $f \in A_\tau \subset \mathbb{Z}[x_1, \ldots, x_\tau]$, with $deg(f) = n$ and $f = \sum_{i=0}^{n} p_i(x_1, \ldots, x_{\tau-1}) x_\tau^i$, where $p_i \in \mathbb{Z}[x_1, \ldots, x_{\tau-1}]$ for $0 \leq i \leq n$ and $p_n \neq 0$. The interesting scenario is, when $p_n = p_{n-1} = \ldots = p_0 = 0$. This polynomial system needs to be analyzed. If the

solution space turns out to be of positive dimension, then f is not well-oriented, if the dimension is 0, then the finite amount of points must be considered when producing a CAD of $A_\tau$. If the system has no solution in the reals, then nothing needs to be done.
One straight forward way would be to produce a CAD of $\{p_0, \ldots, p_n\}$ to find the solution. This is very well possible and while it appears to be very inefficient at first, this is the same thing that McCallums operator actually does by adding all coefficients to the projection. The improvement however would still be that a separate CAD of $\{p_0, \ldots, p_n\}$ would not deal with the mix of other factors (coefficients, discriminants and resultants from other polynomials) during projection, like McCallums original would have. So this approach is in theory always more efficient than McCallums operator. Another advantage is, that any other method for solving equalities over the reals could be taken to determine the isolated points, since only equalities need to be considered and thus using CAD would may be overkill.
The way how to add the points, where a polynomial is identically zero is described in algorithm 1 and 2. In addition Brown proposed another optimization for the lifting phase, which is motivated by the idea, that delineating polynomials not necessarily have to be added every time, when a polynomial vanishes over a cell. It is only mandatory, when the polynomial was added because of a discriminant. This is due to the lifting theorem only requiring the discriminants to have constant order on the submanifolds. This can be implemented by tagging every polynomial, that was added in the projection phase because of a discriminant of a projection set of greater dimension. This however ruins the guarantee of receiving an order-invariant CAD and only assures sign-invariance, which is perfectly fine for most applications, including SMT-solving. Another benefit from tagging the polynomials derived from discriminants is, that a polynomial, that vanishes over a positive dimensional submanifold will only cause problems, if it is tagged. If it is not tagged, it can be ignored, as only sign-invariance is required, which is certainly given on this cell.
Now pseudo-code is given to show the modifications in the lifting phase.

Preprocessing:
**Data:** $\tau$: dimension of input polynomials. Sets $A_1, \ldots, A_\tau$
**Result:** If input polynomials are not well-oriented: Error, else: sets $Q_i$ where a $i+1$-variate polynomial is identically zero.
**for** $i = 3; i \leq \tau; i = i + 1$ **do**
  **for** $f \in A_i$ *with* $f = \sum_{k=0}^{n} p_k(x_1, \ldots, x_{i-1}) x_\tau^k$ **do**
    **if** $p_1 = \ldots = p_n = 0$ *has infinite solutions* **then**
      **return** *ERROR - Not Well-Oriented*
    **else if** $p_1 = \ldots = p_n$ *has no solution in the reals* **then**
      **do nothing**
    **else**
      **for** $\alpha = (\alpha_1, \ldots, \alpha_{i-1})$ *with* $p_1(\alpha) = \ldots = p_n(\alpha) = 0$ **do**
        $Q_{i-1} = Q_{i-1} \cup \{(x_{i-1} - \alpha_{i-1})\}$
      **end**
    **end**
  **end**
**end**

Algorithm 1: Isolating points to add

Lifting:
**Data:** polynomial $f \in A_i$, sample point $\alpha \in \mathbb{R}^{\tau-1}$
**if** *there is a point* $\beta = (\beta_1, \ldots, \beta_{i+1}) \in Q_{i+1}$ *with* $\alpha_k = \beta_k$ *for all* $1 \leq k \leq i$ **then**
  Lift $\alpha$ with respect to $A_{i+1} \cup \{x_{i+1} - \beta_{i+1}\}$ instead of just $A_{i+1}$
**else if** $f(\alpha, x_i)$ *is identically zero* **then**
  **if** $\alpha$ *represents zero dimensional space* **then**
    **if** $f$ *is tagged* **then**
      replace $f$ by the minimal delineating polynomial $q$ of $f$
    **else**
      **do nothing** ($f$ is sign-invariant on the space $\alpha$ represents)
    **end**
  **else**
    **if** $f$ *is tagged* **then**
      **return** *ERROR - Not Well-Oriented*
    **else**
      **no-op** ($f$ is sign-invariant on the space $\alpha$ represents)
    **end**
  **end**
**else**
  Do the usual lifting
**end**

Algorithm 2: Modified Lifting Stage

The following example, taken from [Bro01a], shows the potential difference, that Browns operator could have compared to McCallum.

**Example 5.4.2.** *A classic scenario in elementary school is, analyzing polynomials of degree 2 with respect to their zeros. CAD can deal with the generic case by inspecting* $f(a,b,c,x) := ax^2 + bx + c$. *Let* $a < b < c < x$ *be the variable order,*$A_4 = \{f\}$.
***CAD using McCallum:***
*Projection phase:*

- $discr_x(f) = b^2 - 4ac := g$.
- $projM(A_4) = \{g,a,b,c\} \rightarrow A_3 := \{g,c\}$. *a and b get pushed down.*
- $discr(g) = discr(c) = 1, res_c(g,c) = b^2$.
- $projM(A_3) = \{1, b^2, -4a, -b^2\} \rightarrow A_2 := \{b\}$. *a gets pushed down.*
- $projM(A_2) = \{a\} \rightarrow A_1 = \{a\}$.

*Base phase:*

- $S_1 = (-\infty,0), S_2 = [0,0], S_3 = (0,\infty)$.
- $\rightarrow CAD(A_1) = \{-1,0,1\}$.

*Lifting phase:*

- $CAD(A_2) = \{(0,0),(0,1),(0,-1),(1,0),(1,1),(1,-1),(-1,0),$ $(-1,1),(-1,-1)\}$.
- *Since* $c \in A_3, CAD(A_3)$ *has at least 27 points.*
- ...

***CAD using Brown:***
*Projection phase:*

- $discr_x(f) = b^2 - 4ac := g$ *(tagged).*
- $projM(A_4) = \{g,a\} \rightarrow A_3 := \{g\}$. *a gets pushed down.*
- $discr_c(g) = 1$.
- $projM(A_3) = \{\} \rightarrow A_2 := \{\}$.
- $\rightarrow A_1 = \{a\}$

*Preprocessing:*

- $g(0,0,c)$ *is identically zero* $\rightarrow Q_2 = \{(0,0)\}$.
- $f(0,0,0,x)$ *is identically zero* $\rightarrow Q_3 = \{(0,0,0)\}$.

*Base phase:*

- $S_1 = (-\infty,0), S_2 = [0,0], S_3 = (0,\infty)$.
- $\rightarrow CAD(A_1) = \{-1,0,1\}$.

*Lifting phase:*

- $Q_2 = \{(0,0)\} \to 0$ *needs to be lifted with respect to* $A_2 \cup \{b - 0\}$.
- $CAD(A_2) = \{(0,0),(0,1),(0,-1),(1,0),(-1,0)\}$.
- $Q_3 = \{(0,0,0)\} \to (0,0)$ *needs to be lifted with respect to* $A_3 \cup \{c - 0\}$.
- $g(0,1) = 1, g(0,-1) = 1, g(1,0) = -4c, g(-1,0) = 4c$.
- $g(0,0) = 0, g$ *is tagged, replace with delineating polynomial:*
  $D_1(g) = \{-4c, 2b, -4a\}, D_1(g(0,0)) = \{-4c, 0\} \to q = c$ *(normalized).*
- $CAD(A_3) = \{(0,1,0),(0,-1,0),(1,0,-1),(1,0,0),(1,0,1),(-1,0,-1),$
  $(-1,0,0),(-1,0,1),(0,0,-1),(0,0,0),(0,0,1)\}$
- ...

# Chapter 6

# Experimental Results

The four Operators presented in this paper got implemented in the SMT-RAT toolbox (See [CKJ+15] for more information), which allows to compose SMT-solvers, by offering a SAT-Solver, theory modules for various applications (including a CAD module based on Browns operator) and interfaces to create own modules. Different theory modules can be combined using strategies. All mathematical operations are implemented in the computer Arithmetic and logic library (cArl), which provides all the necessary functions for the operators.
The test setting was based on the implemented CAD module, which uses some optimizations. Constants get removed instantly and are not projected down. The same also goes for positive definite and negative definite functions, as they are always sign- and order-invariant. Another aspect is, that polynomials always get normalized and thus, only the primitive part of each polynomial is considered. While SMT-RAT also can be run in incremental fashion and thus can deal with backtracking scenarios by offering ways of adding and removing polynomials on each projection level, this feature was not utilized here as the study of interest is the efficiency of the operators by the theoretical improvements, that were made on the field of CAD. Also note, that the CAD module of SMT-RAT projects the polynomials down to the univariate case right when they are added. So it always project single polynomials instead of polynomial sets and projects them in depth-first as opposed to breadth-first fashion.
The Benchmarks are taken from the SMT-COMP and in the first run, the composed solver only takes the received polynomials of each formula and projects them down. The tests were run on AMD Opteron 6172 with a timeout limit of 60 seconds. Table 6.1 shows, how many benchmarks the different solvers were able to run, given this settings. Then the benchmarks, that all 4 solvers were able to project down was taken and analyzed more closely. This benchmarks are exactly all tests, that Collins operator was able to handle.

| Operator | Solved Benchmarks |
|---|---|
| Collins | 5698 |
| Hong | 6052 |
| McCallum | 6828 |
| Brown | 6828 |

Table 6.1: Number of test runs

The performance of the four participants on this tests was observed more closely. The rough structure of this tests can be seen in table 6.2.

| | avg | min | 25% qt | median | 75% qt | max |
|---|---|---|---|---|---|---|
| number of polys | ≈ 6,37 | 1 | 5 | 6 | 8 | 34 |
| max degree(main var) | ≈ 5,23 | 1 | 2 | 3 | 6 | 44 |
| max degree (combined) | ≈ 6,06 | 1 | 2 | 4 | 7 | 44 |

Table 6.2: Benchmark structure (qt=quartile), 5698 instances

On the first four projection levels, the number of polynomials and the maximum degree got tracked, the results are depicted in the tables 6.3,6.4 ,6.5 and 6.6. Only few solved instances caused projections with more then 5 variables, which is why this restriction is made. Noticeable things are, that the upper bound on the degree in the main variable is only slightly different from the one, where the combined degree is considered. The combined degree is the sum over the degrees of each variable in a term. This implies, that the polynomials in this benchmarks have a very simple structure. Also with Browns operator it happened, that a whole projection level could be skipped on 11 instances on projection level 1 and on 137 instances on projection level 2. Despite that, Browns operator could only handle the exact same instances with this 60 second upper bound on runtime, as McCallums was able to deal with. So at least in the projection phase, the improvements of Browns operator are not too big. This however is no big surprise, as the real strength of Browns operator is in the lifting phase, where the amount of lifting points are kept smaller due to the manual adding of points, only when it is necessary. Looking at the statistics that regard polynomial degrees, Collins and Hongs operators scored similar results. The same can be seen, when comparing McCallums operator to the one of Brown. However, there is a gap between this 2 groups (Collins approach vs McCallums design). This indicates the huge step, that McCallums operator was able to make in the context of CAD. The last thing to mention about the benchmarks is, that the maximum amount of polynomials can pretty much explode using Collins operator, while the refinement of Hong was able to keep the number of polynomials much smaller. This also gives an idea about the importance of Hongs operator in practical uses (especially, since Hongs operator can be used on any set of integer polynomials and not just well-oriented ones, unlike McCallums and Browns operators).

| number of polys | avg | min | 25% qt | median | 75% qt | max |
|---|---|---|---|---|---|---|
| Collins | ≈ 10,89 | 1 | 3 | 6 | 16 | 99 |
| Hong | ≈ 8,62 | 1 | 2 | 5 | 12 | 57 |
| McCallum | ≈ 6,06 | 1 | 2 | 4 | 8 | 31 |
| Brown | ≈ 5,29 | 1 | 2 | 4 | 7 | 25 |
| max degree(main var) | avg | min | 25% qt | median | 75% qt | max |
| Collins | ≈ 7,06 | 1 | 2 | 3 | 10 | 182 |
| Hong | ≈ 7,05 | 1 | 2 | 3 | 10 | 182 |
| McCallum | ≈ 6,05 | 1 | 2 | 2 | 8 | 182 |
| Brown | ≈ 6,04 | 1 | 2 | 2 | 8 | 182 |
| max degree(combined | avg | min | 25% qt | median | 75% qt | max |
| Collins | ≈ 7,76 | 1 | 2 | 4 | 10 | 182 |
| Hong | ≈ 7,75 | 1 | 2 | 4 | 10 | 182 |
| McCallum | ≈ 6,68 | 1 | 2 | 3 | 8 | 182 |
| Brown | ≈ 6,68 | 1 | 2 | 3 | 8 | 182 |

Table 6.3: Results on projection level 1 (5693 instances)

| number of polys | avg | min | 25% qt | median | 75% qt | max |
|---|---|---|---|---|---|---|
| Collins | ≈ 783,14 | 1 | 2 | 11 | 116 | 24489 |
| Hong | ≈ 158,77 | 1 | 2 | 6 | 43 | 6056 |
| McCallum | ≈ 16,74 | 1 | 2 | 5 | 15 | 227 |
| Brown | ≈ 11,62 | 1 | 2 | 4 | 12 | 168 |
| max degree(main var) | avg | min | 25% qt | median | 75% qt | max |
| Collins | ≈ 26,32 | 1 | 1 | 6 | 26 | 419 |
| Hong | ≈ 26,16 | 1 | 1 | 6 | 26 | 419 |
| McCallum | ≈ 13,29 | 1 | 1 | 3 | 16 | 264 |
| Brown | ≈ 13,48 | 1 | 1 | 3 | 16 | 264 |
| max degree(combined) | avg | min | 25% qt | median | 75% qt | max |
| Collins | ≈ 26,39 | 1 | 1 | 6 | 26 | 419 |
| Hong | ≈ 26,23 | 1 | 1 | 6 | 26 | 419 |
| McCallum | ≈ 13,34 | 1 | 1 | 3 | 16 | 264 |
| Brown | ≈ 13,53 | 1 | 1 | 3 | 16 | 264 |

Table 6.4: Results on projection level 2 (5583 instances)

| number of polys | avg | min | 25% qt | median | 75% qt | max |
|---|---|---|---|---|---|---|
| Collins | ≈ 117,00 | 1 | 1 | 3 | 10 | 22806 |
| Hong | ≈ 20,20 | 1 | 1 | 3 | 9 | 2009 |
| McCallum | ≈ 5,06 | 1 | 1 | 2 | 7 | 80 |
| Brown | ≈ 4,65 | 1 | 1 | 2 | 7 | 73 |
| max degree(main var) | avg | min | 25% qt | median | 75% qt | max |
| Collins | ≈ 11,79 | 1 | 1 | 1 | 4 | 286 |
| Hong | ≈ 11,66 | 1 | 1 | 1 | 4 | 286 |
| McCallum | ≈ 5,25 | 1 | 1 | 1 | 4 | 64 |
| Brown | ≈ 5,03 | 1 | 1 | 1 | 4 | 64 |
| max degree(combined) | avg | min | 25% qt | median | 75% qt | max |
| Collins | ≈ 11,82 | 1 | 1 | 1 | 4 | 286 |
| Hong | ≈ 11,68 | 1 | 1 | 1 | 4 | 286 |
| McCallum | ≈ 5,28 | 1 | 1 | 1 | 4 | 64 |
| Brown | ≈ 5,06 | 1 | 1 | 1 | 4 | 64 |

Table 6.5: Results on projection level 3 (789 instances)

| number of polys | avg | min | 25% qt | median | 75% qt | max |
|---|---|---|---|---|---|---|
| Collins | ≈ 15,61 | 1 | 2 | 4 | 11 | 302 |
| Hong | ≈ 10,27 | 1 | 2 | 4 | 8 | 106 |
| McCallum | ≈ 7,91 | 1 | 2 | 3 | 6 | 29 |
| Brown | ≈ 5,46 | 1 | 2 | 3 | 5 | 26 |
| max degree(main var) | avg | min | 25% qt | median | 75% qt | max |
| Collins | ≈ 4,02 | 1 | 1 | 2 | 5 | 32 |
| Hong | ≈ 3,84 | 1 | 1 | 2 | 5 | 24 |
| McCallum | ≈ 3,43 | 1 | 1 | 1 | 4 | 16 |
| Brown | ≈ 2,74 | 1 | 1 | 1 | 4 | 16 |
| max degree(combined) | avg | min | 25% qt | median | 75% qt | max |
| Collins | ≈ 5,25 | 1 | 1 | 2 | 6 | 32 |
| Hong | ≈ 5,07 | 1 | 1 | 2 | 6 | 24 |
| McCallum | ≈ 3,84 | 1 | 1 | 2 | 4 | 16 |
| Brown | ≈ 3,48 | 1 | 1 | 2 | 4 | 16 |

Table 6.6: Results on projection level 4 (147 instances)

This statistics are also visualized in the form of boxplots in the appendix. The boxplots were made with pgfpfots and have the following build rule. Let the inter-quantile-range (IQR) denote the number given by the 75% quartile minus the number that the 25% quartile denots. The 25% quartile denotes the smallest data point, that is greater than 25% of the data. In symmetric fashion, the 75% is the greatest data point, that is smaller than 75% of all data points. The lower (left) whisker is set to smallest number, that is greater than the 25% quartile - $1.5 \cdot IQR$ and the upper whisker: biggest number, that is smaller than the 75% quartile + $1.5 \cdot IQR$. The bar inside the box represents the median and black points illustrate the data points, that are outliers with respect to the boxplots. The range depicted in these figures is chosen, such that the maximum value is always visible. In most cases this required some non-continuity on the $x$-axis.

In a second test run, the 5698 samples were given to the full SMT-solver. The time bound was again set to 60 seconds and the following table shows the success rate of this procedure using the different operators.

| porjection operator | solved formulas | timeouts | timeut relative |
|---|---|---|---|
| Collins | 5041 | 657 | $\approx$ 11,53% |
| Hong | 5125 | 573 | $\approx$ 10,06% |
| McCallum | 5284 | 414 | $\approx$ 7,27% |
| Brown | 5299 | 399 | $\approx$ 7,00% |

Table 6.7: General statistics of full SMT-procedure

This rate of timeouts is pretty low, considering the given setting. The benchmark set was chosen, so that the raw projection could be done in 60 seconds by every operator. Collins operator was only able to exactly handle those benchmarks. Now in the full SMT-solving settings, one could expect higher time consumption. But SMT-RAT is very efficient, when it comes to keeping the projection size as small as possible. Close analysis within the SAT-module enables SMT-RAT to filter out some polynomials based on the boolean structure. In fact, some of the benchmarks, that caused the greatest amount of polynomials in the first test run, turned out to be solvable in a very short amount of time. One example is the benchmark "meti-tarski/polypaver-bench-sqrt-3d-chunk-0479.smt2". It had a total amount of 24575 polynomials during the projection phase in test phase one. It could be solved in 33 milliseconds, due to some contradicting bounds.
Also the full SMT-solver of SMT-RAT creates partial CADs and checks for satisfiability. So it could happen, that satisfiable formulas can be solved before the whole projection is done. The runtimes on the benchmarks, that could be finished is given in table 6.8.

| runtime (ms) | avg | min | 25% qt | median | 75% qt | max |
|---|---|---|---|---|---|---|
| Collins | $\approx$ 452,80 | 16 | 27 | 32 | 47 | 56538 |
| Hong | $\approx$ 233,30 | 17 | 28 | 36 | 55 | 48504 |
| McCallum | $\approx$ 216,38 | 20 | 34 | 45 | 72 | 38727 |
| Brown | $\approx$ 220,11 | 17 | 27 | 35 | 52 | 49497 |

Table 6.8: runtimes on the finished benchmarks

An interesting note to mention is, that according to the data, McCallum was faster in average than Brown. This partly is because usnig McCallums operator, the solver could not solve some harder benchmarks, which the use of Browns operator allowed to solve. The scatter plot in 6.1 shows the competition of these operators on the set of benchmarks, that both solvers could solve in time:

The expected behavior based on the theoretical superiority of Browns operator over McCallums operator is not reflected by the benchmarks. This can be explained by the way, that the SMT-solver works. Since partial CADs are used to check for satisfiability before the full projection is done. If the coefficients that McCallums operator includes and Browns does not include get projected first and this partial projection is sufficient to find sample points to satisfy the formula, then McCallums

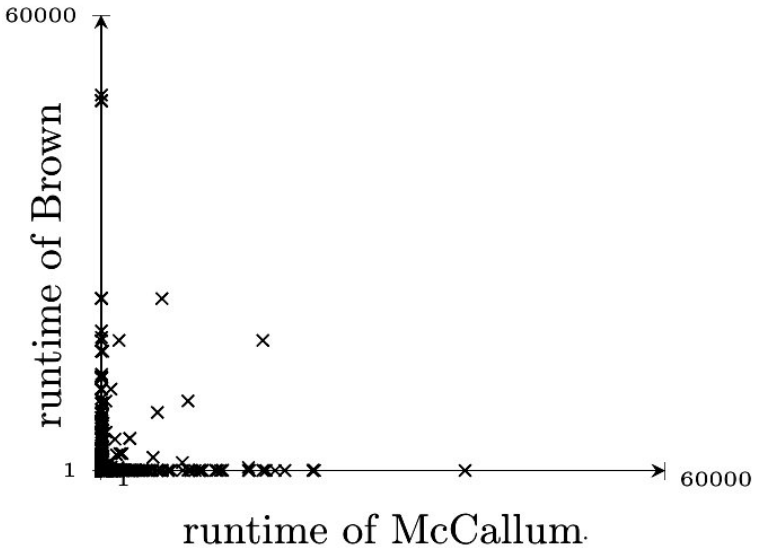


Figure 6.1: runtimes Brown vs McCallum

operator is faster. This is also due to the degree, which on the used benchmarks does shrink on the coefficients, while the degree of discriminants and resultants is likely to grow. So a partial CAD based on the coefficients results in less sample points and can return faster answers.

# Chapter 7

# Conclusion

This thesis gave an overview about different approaches to the projection operator in the CAD procedure. The main theoretical results, that are necessary to validate the presented operators, got collected and all extra requirements in order to use those operators in the context of SMT-solving were highlighted. The practical analysis using SMT-RAT mirrored the theoretical results on the number of projected polynomials. Future work might include a closer look at the side cases, when using Browns projection on examples, where points need to be added to the projection. Also the runtimes on the tests, where Browns operator competed versus McCallums operator, suggest a dual approach, where McCallums operator is used for a bounded time. If satisfiability can be determined, it is returned, else one could switch back to using Browns operator to complete the projection on harder cases, where the theoretical superiority allows a faster lifting as the projection set grows. Some further improvements can also be made in the projection phase, when dealing with bounded sets as mentioned in the second part of[Bro01a]. Another quite promising approach is to use equational constraints to simplify the CAD procedure, see [BDE+14].

# Bibliography


[ACM82] Dennis S. Arnon, George E. Collins, and Scott McCallum. Cylindrical algebraic decomposition i: The basic algorithm. Technical report, Purdue University, Department of Computer Science Technical Reports, 1982.

[ACM84] Dennis S. Arnon, George E. Collins, and Scott McCallum. Cylindrical algebraic decomposition i: The basic algorithm. *SIAM J. Comput.*, 13(4):865–877, November 1984.

[BDE+14] Russell Bradford, James H. Davenport, Matthew England, Scott McCallum, and David Wilson. Truth table invariant cylindrical algebraic decomposition. *arXiv:1401.0645*, pages 1425–1426, 2014.

[Bro01a] Chistopher W. Brown. Improved projection for cylindrical algebraic decomposition. *Journal of Symbolic Computation*, 32:447–465, 2001.

[Bro01b] Christopher W. Brown. The mccallum projection, lifting, and order-invariance. Technical report, U.S. Naval Academy, Computer Science Department, 2001.

[CKJ+15] Florian Corzilius, Gereon Kremer, Sebastian Junges, Stefan Schupp, and Erika Ábrahám. *SMT-RAT: An Open Source C++ Toolbox for Strategic and Parallel SMT Solving*, pages 360–368. Springer International Publishing, Cham, 2015.

[Col67] George E. Collins. Subresultants and reduced polynomial remainder sequences. *J. ACM*, 14(1):128–142, January 1967.

[Col75] George E. Collins. *Quantifier elimination for real closed fields by cylindrical algebraic decompostion*, pages 134–183. Springer Berlin Heidelberg, Berlin, Heidelberg, 1975.

[GCL92] Keith O. Geddes, Stephen R. Czapor, and George Labahn. *Polynomial Factorization*, pages 337–388. Springer US, New York, 1992.

[Hon90] Hoon Hong. An improvement of the projection operator in cylindrical algebraic decomposition. *ISSAC '90 Proceedings of the international symposium on Symbolic and algebraic computation*, pages 261–264, 1990.

[Jud16] Thomas W. Judson. *Abstract Algebra - Theory and Applications*. Orthogonal Publishing L3c, 2016.

[McC85] Scott McCallum. An improved projection operation for cylindrical algebraic decomposition. Technical report, University of Wisconsin Madison, 1985.

[Tar48] Alfred Tarski. A decision method for elementary algebra and geometry. Technical report, University of California Press, 1948.

[Zar75] Oscar Zariski. On equimultiple subvarieties of algebroid hypersurfaces. *PNAS 'Proceedings of the National Academy of Sciences of the United States of America*, pages 1425–1426, 1975.

# Appendix A

# Experimental Results

Figure A.1: Level 0 (before projection): 5698 data points

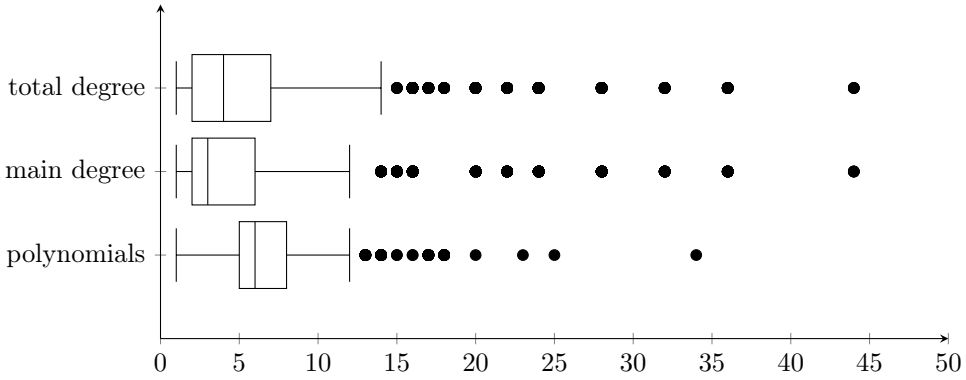


Figure A.2: number of polynomials in projection level 1 (5693 data points)

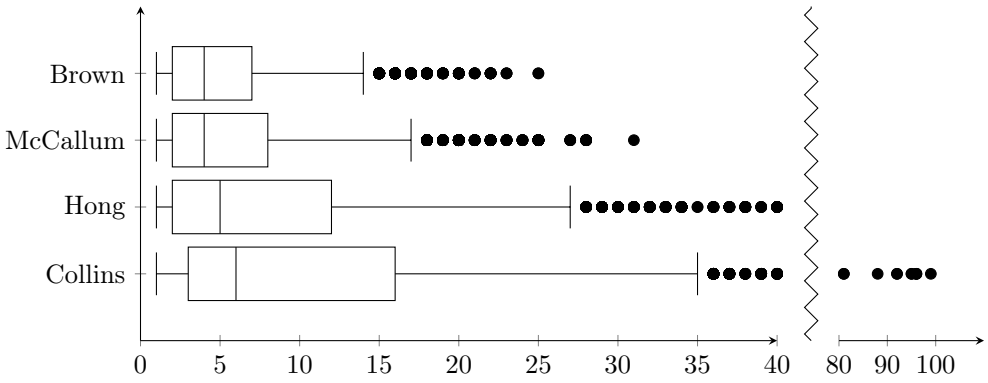

Figure A.3: degree in main variable in projection level 1 (5693 data points)

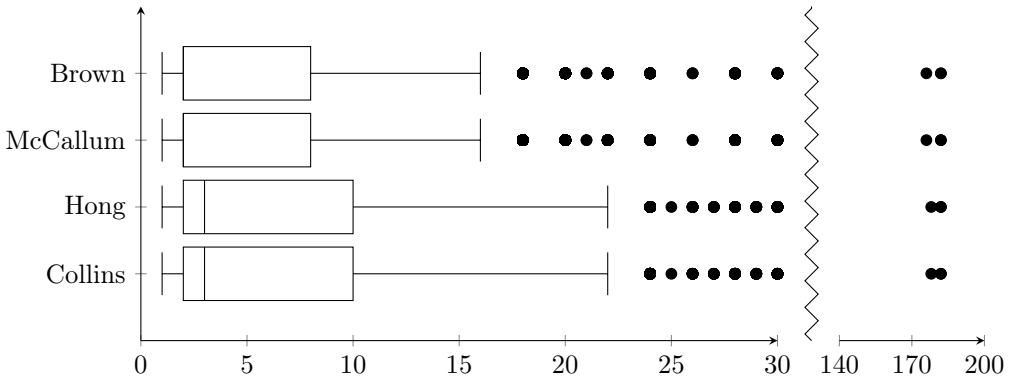


Figure A.4: combined degree in projection level 2 (5693 data points)

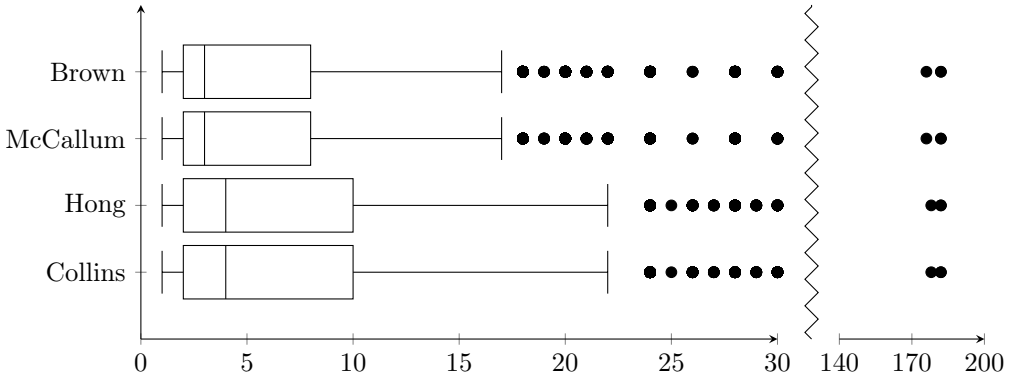


Figure A.5: number of polynomials in projection level 2 (5583 data points)

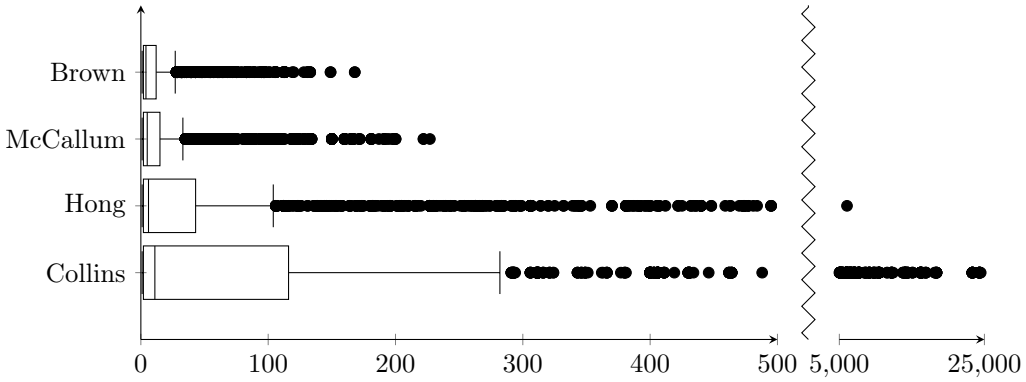

Figure A.6: degree in main variable in projection level 2 (5583 data points)

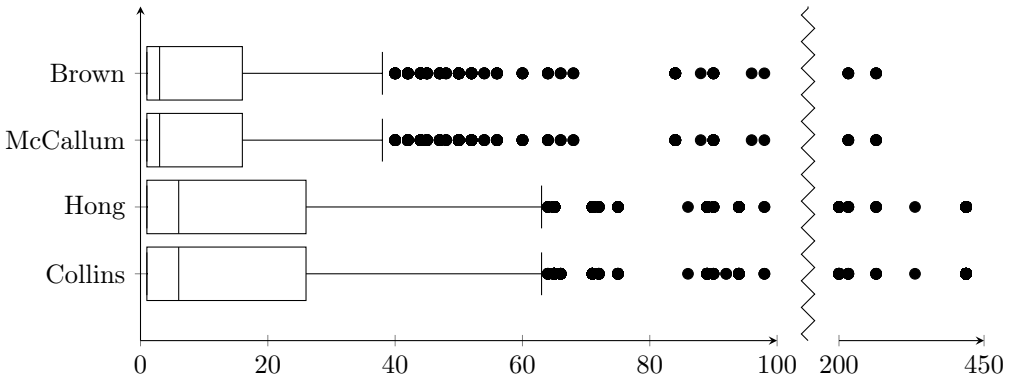


Figure A.7: combined degree in projection level 2 (5583 data points)

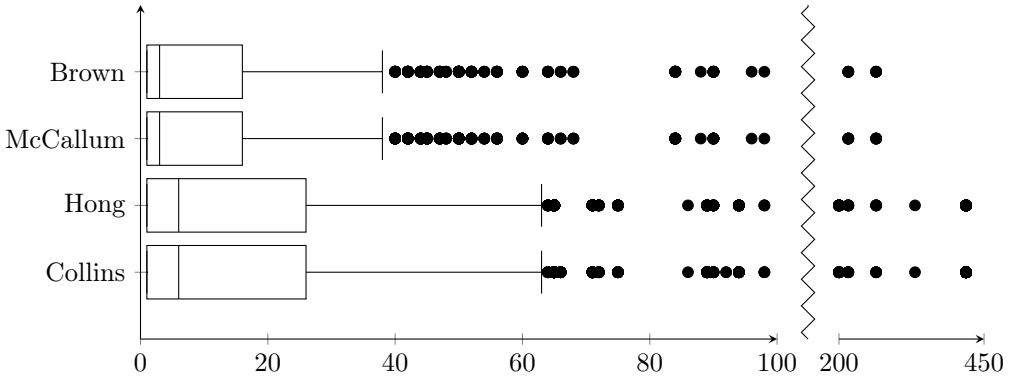


Figure A.8: number of polynomials in projection level 3 (789 data points)

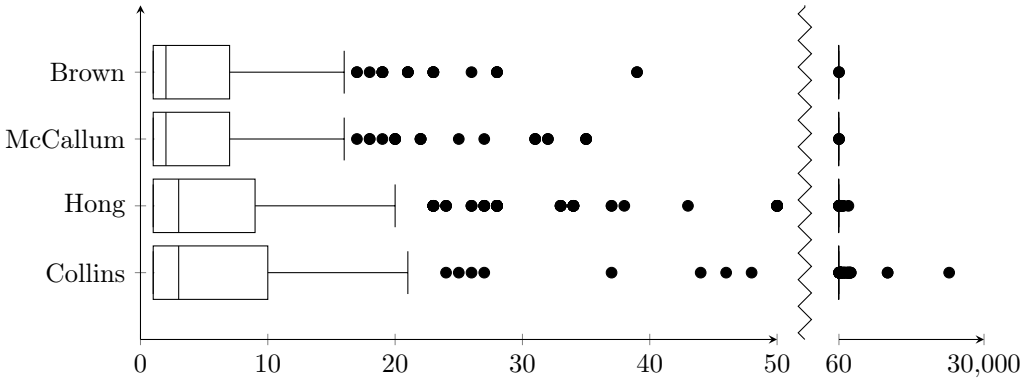

Figure A.9: degree in main variable in projection level 3 (789 data points)

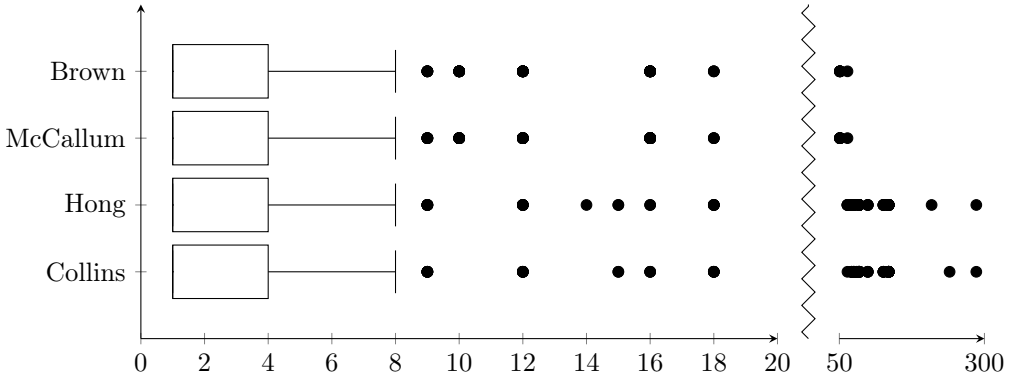


Figure A.10: combined degree in projection level 3 (789 data points)

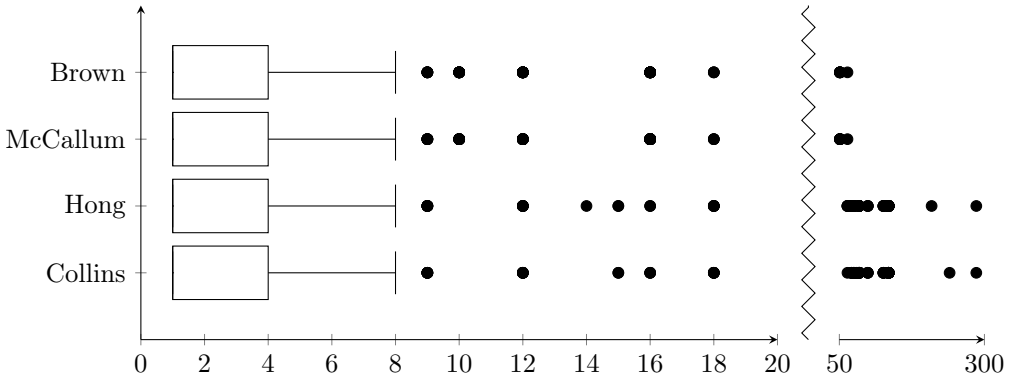


Figure A.11: number of polynomials in projection level 4 (147 data points)

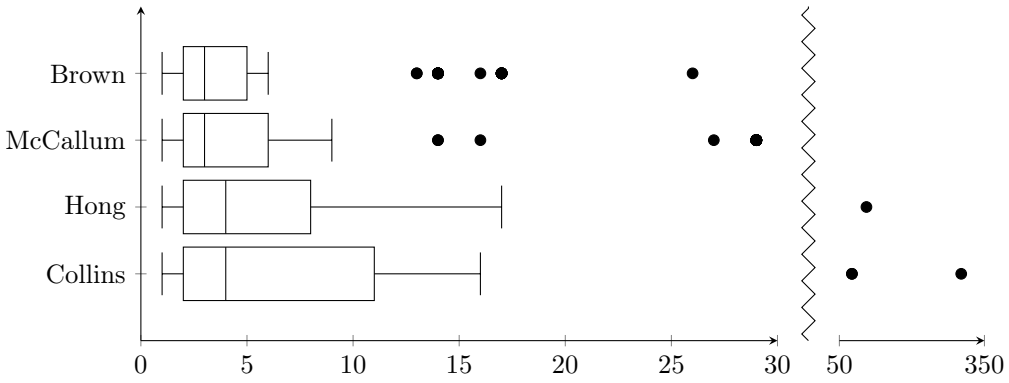

Figure A.12: degree in main variable in projection level 4 (147 data points)

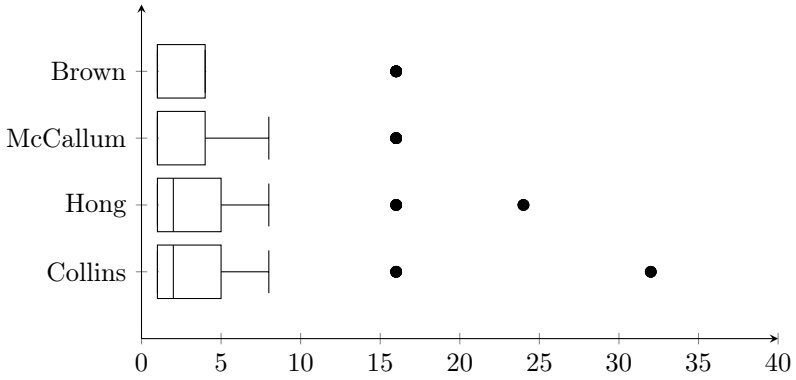


Figure A.13: combined degree in projection level 4 (147 data points)