Variable Elimination from Non-linear Boolean Equation Systems

Bjørn Møller Greve[a,b,*], Håvard Raddum[b], Gunnar Fløystad[c], Øyvind Ytrehus[b]

[a] *Norwegian Defence Research Establishment*
[b] *Simula@UiB*
[c] *Dept. of Mathematics, Universitetet i Bergen, Norway*

Abstract

We study Boolean equation systems, and how to eliminate variables from them while bounding the degree of polynomials produced during elimination. Tools used for eliminating variables are explained, a procedure for variable elimination is introduced, and we relate our techniques to Gröbner bases and XL methods. Finally we show that the tools and elimination procedure developed result in a more refined and efficient elimination algorithm than multiplying all polynomials with a set of monomials and doing Gaussian elimination on the resulting Macaulay matrix.

Key words: Systems of Boolean equations, Elimination of variables, Syzygies, Gröbner bases, XL, degree bound.

1. Introduction

In this paper we consider non-linear systems of Boolean equations

$$\begin{array}{rcl} f_1(x_0,\ldots,x_n) & = & 0 \\ f_2(x_0,\ldots,x_n) & = & 0 \\ \vdots & & \\ f_m(x_0,\ldots,x_n) & = & 0 \end{array} \tag{1}$$

where each f_i is of bounded degree. How to solve such systems has been studied extensively before, and it is well known that the problem of solving (1) is NP-hard in the general case.

Several methods for solving systems of Boolean equations have been suggested, based on various approaches. In [6] and [7] the authors introduce the XL and XSL algorithms, respectively. The basic idea in these papers is to multiply equations with enough monomials to re-linearize the whole system of Boolean equations. The approach in our paper has similarities to this, but we bound the degree at $\leq d$ for some d in order to control the complexity, and we do not multiply all the f_i with all monomials. Our approach for solving is focused on *variable elimination*, which is not the approach taken in the XL and XSL algorithms. A system of polynomial equations in relatively few variables can be solved by simple exhaustive search. In this paper we try to extract such systems, by investigating how to eliminate variables from a Boolean equation system while working only with polynomials of degree $\leq d$.

Algorithms for computing Gröbner bases, such as Buchberger's algorithm or its improved versions F4/F5 [10, 11] do not limit the degree of polynomials computed during execution. Our approach therefore differs from these in that sense. The main contributions of our work are the introduction of elimination tools that respect a degree bound, a procedure for eliminating variables

*Corresponding author
Email addresses: Bjorn.Greve@uib.no (Bjørn Møller Greve), haavardr@simula.no (Håvard Raddum), Gunnar.Floystad@uib.no (Gunnar Fløystad), oyvindy@simula.no (Øyvind Ytrehus)

in a way that is more efficient than the Gröbner base approach, and the proofs of correctness for this procedure.

The paper is organized as follows. In Section 2 we introduce the notation used and some preliminary results. In Section 3 we introduce *normalization, resultants, coefficient constraints* and *syzygies*, the basic tools and concepts we use for variable elimination. In Section 4 we combine the tools to construct a procedure (algorithm) for variable elimination, and show that the output of the algorithm will be the same as if we had multiplied all polynomials with *all* monomials of prescribed degree, and performed Gaussian elimination on the monomials depending on the variable to be eliminated. In that sense our procedure produces the same output as, but is a lot faster than, the brute force approach of multiplying all polynomials with all monomials.

2. Notation and preliminaries

Consider the quotient ring of Boolean polynomials in n variables. We denote the ring by

$$\mathbb{B}[x_0, \ldots, x_n] = \mathbb{F}_2[x_0, \ldots, x_n]/(x_i^2 + x_i | i = 0, \ldots, n).$$

A monomial is a product $x_{i_1} \cdots x_{i_d}$ of d distinct (because $x^2 = x$) variables, where d is the degree of this monomial. The degree of a polynomial

$$p = \sum_i m_i$$

where the m_i's are distinct monomials, is the maximum degree over the monomials in p. Given a set of polynomial equations

$$F = \{f_i(x_0, \ldots, x_n) = 0 | i = 1, \ldots, m\},$$

our objective is to find its set of solutions in the space $\mathbb{F}_2^n$. The approach we take in this paper is to solve the system of equations by eliminating variables.

Consider the projection which omits the first coordinate:

$$\pi_0 : \mathbb{F}_2^{n+1} \to \mathbb{F}_2^n$$
$$(a_0, a_1, \ldots, a_n) \mapsto (a_1, \ldots, a_n),$$

and denote by $\mathbb{B}[x_1, \ldots, x_n]$ the ring of Boolean polynomials where x_0 has been omitted. We may in a similar fashion consider a sequence of k projections

$$\mathbb{F}_2^{n+1} \to \mathbb{F}_2^n \to \cdots \to \mathbb{F}_2^{n-k+1},$$

where the i'th projection is denoted $\pi_i : \mathbb{F}_2^{n-i+1} \to \mathbb{F}_2^{n-i}$ for $0 \leq i < k$.

2.1. Systems of Boolean equations and ideals

Any Boolean function $f : \mathbb{F}_2^{n+1} \to \mathbb{F}_2$ can be written as a polynomial in the ring $\mathbb{B}[x_0, \ldots, x_n]$ and this is a one-to-one correspondence. Such a function is uniquely determined by the zero set

$$Z(f) = \{\mathbf{a} \in \mathbb{F}_2^{n+1} | f(\mathbf{a}) = 0\}.$$

Conversely, for any given subset Z of $\mathbb{F}_2^{n+1}$ there is a unique Boolean function with this as zero set. So there are one-to-one correspondences between Boolean functions, Boolean polynomials, and subsets of $\mathbb{F}_2^{n+1}$.

Lemma 1. *Let f and g be Boolean polynomials. Then:*

$$f \text{ is a multiple } hg \text{ of } g \Leftrightarrow \text{ the zero sets } Z(f) \supseteq Z(g).$$

Proof. The implication $\Rightarrow$ is clear. Suppose $Z(f) \supseteq Z(g)$ and let H be the difference set. It corresponds then to Boolean polynomial h with zero set $Z(h) = H$. Then f and hg have the same zero set, and hence are equal. ■

On the other hand, for $f \in \mathbb{B}[x_0, \ldots, x_n]$ we may write f on the form $f = \sum_{\alpha \in A} \prod_{i=0}^{n} x_i^{\alpha_i}$, where $\alpha = (\alpha_0, \ldots, \alpha_n)$ is a binary vector of length $n+1$ and $A \subset GF(2)^{n+1}$ specifies the monomials occurring in f. Any Boolean function can be written in this form, which gives a different one-to-one correspondence between Boolean functions and subsets of $\mathbb{F}_2^{n+1}$. This establishes the following relationship

$$A \leftrightarrow f \leftrightarrow Z.$$

For a given function f, the relationship between the sets A and Z is however far from trivial.

For a set of polynomial equations (1), let $F = \{f_1, \ldots, f_m\}$. The polynomials in F generate an ideal $I = (f_1, \ldots, f_m) = I(F)$ in the ring $\mathbb{B}[x_0, \ldots, x_n]$. Let $Z(I)$ denote the zero set of this ideal, i.e, the set of points

$$Z(I) = \{\mathbf{a} \in \mathbb{F}_2^{n+1} | f(\mathbf{a}) = 0 \text{ for every } f \in I\}.$$

Lemma 2. *Let f, g be Boolean functions in $\mathbb{B}[x_0, \ldots, x_n]$. Then the following ideals are equal:*

$$(f, g) = (fg + f + g).$$

Proof. Note that it is easy to verify that they have the same zero set. But it is not a priori clear that they are equal as ideals of Boolean polynomials. Clearly $(f, g) \supseteq (fg + f + g)$. We also have $Z(f, g) = Z(fg + f + g)$, since it is easy to check that $f(a) = g(a) = 0$ if and only if $f(a)g(a) + f(a) + g(a) = 0$. Thus the zero set $Z(f) \supseteq Z(fg + f + g)$. By Lemma 1 the Boolean function f is a multiple $h(fg + f + g)$ for some other Boolean function h, and similarly for g. Thus

$$(f, g) \supseteq (fg + f + g) \supseteq (f, g),$$

which shows that these ideals are equal. ■

Corollary 3. *Any ideal $I = (f_1, \ldots, f_m)$ in $\mathbb{B}[x_0, \ldots, x_n]$ is a principal ideal. More precisely $I = (f)$ where*

$$f = 1 + \prod_{i=1}^{m} (f_i + 1).$$

Proof. Let $I = (f_1, \ldots, f_m)$. By Lemma 2 this is equal to the ideal $(f_1 f_2 + f_1 + f_2, f_3, \ldots, f_m)$, with one generator less. We may continue the process for the remaining generators providing us in the end with $I = (f)$, where

$$f = 1 + \prod_{i=1}^{m} (f_i + 1).$$

■

Corollary 4. *For two ideals in $\mathbb{B}[x_0, \ldots, x_n]$ we have $I \supseteq J$ if and only if $Z(I) \subseteq Z(J)$. In particular $I = J$ if and only if $Z(I) = Z(J)$.*

Proof. By Corollary 3 we have $I = (f)$ and $J = (g)$, where f and g are the respective principal generators. Clearly if $(f) \supseteq (g)$ then $Z(g) \supseteq Z(f)$. If the zero set of g contains the zero set of f, then $g = fh$ for some polynomial h. Hence $(f) \supseteq (g)$. ■

Now given an ideal $I \subset \mathbb{B}[x_0, \ldots, x_n]$, our aim is to find the ideal $I_1 \subset \mathbb{B}[x_1, \ldots, x_n]$ such that $Z(I_1) = \pi_0(Z(I))$. More generally, when eliminating more variables we aim to find the ideal $I_k \subset \mathbb{B}[x_k, \ldots, x_n]$, such that $Z(I_k) = \pi_{k-1} \circ (\cdots \circ (\pi_0(Z(I))))$. Since the degree of the polynomials can grow very quickly when eliminating variables, in practice we have to settle for computing an ideal J, as large as possible given computational restrictions, which is contained in I_k.

Let us first describe precisely the ideal I_k whose zero set is the sequence of projections $\pi_{k-1} \circ (\cdots \circ (\pi_0(Z(I))))$. This corresponds to what is known as the *elimination ideal* $I \cap \mathbb{B}[x_k, \ldots, x_n]$.

Lemma 5. *Let $I_k \subseteq \mathbb{B}[x_k, \ldots, x_n]$ be the ideal of all Boolean functions vanishing on $\pi_{k-1} \circ (\cdots \circ (\pi_0(Z(I))))$. Then $I_k = I \cap \mathbb{B}[x_k, \ldots, x_n]$.*

Proof. We show this for the case when eliminating one variable, the general case follows in a similar manner. Clearly $I_1 \supseteq I \cap \mathbb{B}[x_1, \ldots, x_n]$. Conversely let $f \in \mathbb{B}[x_1, \ldots, x_n]$ vanish on $\pi_0(Z(I))$. Then f must also vanish on $Z(I)$, where f is regarded as a member of the extended ring $\mathbb{B}[x_0, \ldots, x_n]$. Therefore $f \in I$ by Corollary 4. ∎

A standard technique for computing elimination ideals is to use Gröbner bases, which eliminate one *monomial* at the time. In fact, to compute elimination ideals via Gröbner bases one has to compute the full Gröbner basis before performing elimination. Computing Gröbner bases is computationally heavy because the degrees of the polynomials grow rapidly over the iterations. To deal with this problem we propose new tools which also restrict the degree of the polynomials.

Our solution is to not use all polynomials during elimination, but only compute with those that do not produce new polynomials of high degree. We denote an ideal where the degree is restricted to some d by J^d, whereas J^∞ means that we allow all degrees. The benefit from our approach is that the elimination process has much lower complexity, at the cost of the following disadvantage.

Discarding polynomials of degree $> d$ gives an ideal J^d that is only contained in the elimination ideal $J^\infty = I \cap B[k, n]$. It follows that $Z(J^d)$ of the eliminated system contains all the projected solutions of the original set of equations, but it will also contain "false" solutions which will not fit the ideal I when lifted back to $\mathbb{F}_2^{n+1}$, regardless of which values we assign to the eliminated variables. Since the proposed procedure expands the solution space to include false solutions, the worst case scenario is when we end up with an empty set of polynomials after eliminating a sequence of variables. This means that all constraints given by the initial I have been removed, and we end up with the complete $\mathbb{F}_2^{n-k+1}$ as a solution space.

It is important to note that not discarding any polynomials will provide, by Lemma 5, only the true solutions to the set where variables have been eliminated. Hence the solutions can then be lifted back to the solutions of the initial ideal I. The drawback of this approach is that we must be able to handle arbitrarily large polynomials, i.e high computational complexity.

Thus there is a tradeoff between the maximum degree d allowed, and the proximity between the "practical" ideal J^d and the true elimination ideal J^∞. In the following we are going to dig deeper into this trade-off.

In the remainder of the paper we adopt the following non-standard notation: For a polynomial f^i, the superscript i indicates that f has degree $\leq i$, and does not mean f raised to the power i. The reason for this is that since we bound the degree of polynomials we work with, it is important to keep in mind which degree the various polynomials and monomials have. Therefore we indicate this with a superscript. Similarly, F^i for a set of polynomials indicates that all polynomials in F have degree $\leq i$.

We can split the initial system F into d sets according to degree:

$$F^d = \{f_1^d, \ldots, f_{r_d}^d\}, F^{d-1} = \{f_1^{d-1}, \ldots, f_{r_{d-1}}^{d-1}\}, \ldots, F^2 = \{f_1^2, \ldots, f_{r_2}^2\}, F^1 = \{f_1^1, \ldots, f_{r_1}^1\},$$

The polynomials in $F^d, F^{d-1}, \ldots, F^1$ together generate the ideal $I = (F^d, F^{d-1}, \ldots, F^1)$. The focus in this paper is to eliminate variables from a system of Boolean equations. From here on we therefore always assume that $F^1 = \emptyset$ and omit it from further analysis, since otherwise we could

just use any (linear) polynomial in F^1 to eliminate a variable without changing the degree of the system.

For a set of polynomials F, we use the notation F_{x_0} to mean that all polynomials in F depend on the variable x_0. Similarly, $F_{\overline{x_0}}$ indicates that x_0 does not appear in any of the polynomials in F.

3. Elimination Techniques

Our approach to solve the system (1) is to eliminate variables so that we find degree $\leq d$ polynomials in I_k, in smaller and smaller Boolean rings $\mathbb{B}[x_k, \ldots, x_n]$. Our objective is to find as many polynomials in the ideal I generated by $F^d, F^{d-1}, \ldots, F^2$ as possible *computing only with polynomials of degrees* $\leq d$. This limits both storage and computational complexity.

Let $F = (f_1, \ldots, f_m)$ be the set of Boolean equations in $\mathbb{B}[x_0, \ldots, x_n]$ of degree $\leq d$, and denote by $\langle F \rangle$ the vector space spanned by the polynomials in F, where each monomial is regarded as a coordinate. Let $L = \{1, x_0, \ldots, x_n\}$, such that $\langle L \rangle$ is the vector space spanned by the Boolean polynomials of degree ≤ 1. In order to enable us to multiply with higher degree monomials, we generalize L as follows. Set $L^0 = \{1\}$ and $L^1 = \{1, x_0, \ldots, x_n\}$. With this notation we set L^2 to be the monomials of degree ≤ 2. More generally, we define L^i to be the set of monomials of degree $\leq i$, so it is actually the i'th power of L. For the set of polynomials F^i, $(i = 2, \ldots, d-1)$ we consider the set $L^j F^i$ of all products lf^i where $l \in L^j$ and $f^i \in F^i$. Since we are bounding the maximal degree to be d, we can be certain to form any such product provided that $i + j \leq d$. With this constraint, the total set of polynomials we can construct in our analysis is the vector space generated from the following basis:

$$F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \cdots \cup L^{d-2} F^2.$$

For the purpose of this paper we shall need two types of total orders on the monomials in a Boolean ring.

1. An x_0-elimination order: If t_1 and t_2 are monomials where t_1 contains x_0 while t_2 does not, then $t_1 > t_2$. An example is the lexicographic order.

2. A degree order: If t_1 has degree larger than t_2 then $t_1 > t_2$.

As part of the variable elimination process it will be necessary to split a set of polynomials according to dependency on x_0 and possibly according to degree. The procedures for this can be implemented in terms of row reduction on the *Macaulay matrix* of the given set: Its columns are indexed by the monomials according to the total order we use, and the rows are indexed by the polynomials in our set. The entries are the coefficients of the monomials of the polynomials. A basic procedure is:

$SplitVariable(F, x_0)$: We use an x_0-elimination order. We perform Gaussian elimination on the Macaualy matrix of F on all the columns indexed by monomials containing x_0 (which form an initial segment of columns) to get these columns in row-reduced echelon form. The rest of the columns are not of concern in this procedure. The outputs are sets of polynomials F_{x_0} consisting of polynomials which containing x_0-terms (the upper rows of the resulting matrix), and $F_{\overline{x_0}}$ consisting of those polynomials not containing x_0-terms (the remaining lower rows of the resulting matrix).

3.1. Normalization

The purpose of normalization is to eliminate many monomials containing x_0 from a given polynomial, using a set of lower-degree polynomials depending on x_0 as a basis. Suppose we have an x_0-elimination ordering. Let $f^i \in F^i_{x_0}$ be given, and let $G_{x_0} = \{g_1, \ldots, g_r\}$ be a set of polynomials of degree $\leq i$ which all contain x_0. In general we say that f^i is normalized with respect

to G_{x_0} if no term in f^i is divisible by any leading term of the polynomials in G_{x_0}, and we write $f^{i,norm}$ when we need to stress that f^i is normalized (with respect to some basis).

Without restriction on the degree, it is easy to create $f^{i,norm}$ from f^i and G_{x_0}. Simply run through the polynomials in G_{x_0} and check if some monomial m_f in f^i is divisible by a leading term of some $g_j \in G_{x_0}$. If $m_f = q * in(g_j)$, eliminate m_f by adding qg_j to f^i. Doing this successively for all polynomials in G_{x_0} will produce $f^{i,norm}$.

Since we need to restrict the degree of the polynomials we work with to be at most d, extra care has to be taken. The leading term $in(g_j)$ of some g_j may not be of the highest degree among the monomials in g_j, for instance if $g_j = x_0x_3 + x_1x_3x_4$. When this happens we can only eliminate the m_f in f^i where $deg(q) + deg(g_j) \leq d$, since we do not want to possibly introduce terms of degree larger than d. We therefore give the following definition of what normalization means in this paper.

Definition 6. *Let $f^i \in F^i_{x_0}$ and $d \geq i$ be given, and let G_{x_0} be a set of polynomials all depending on x_0 with an x_0-elimination ordering. We say that f^i is* normalized *with respect to G_{x_0} if no term m_f in f^i can be written as $m_f = q * in(g)$ for any $g \in G_{x_0}$ where $deg(q) + deg(g) \leq d$. When writing $F^{i,norm}$ for a set of polynomials we mean that every polynomial in F^i is normalized with respect to some basis, and that all the polynomials have distinct initial terms.*

In our algorithm we combine normalization and Gaussian reduction to get polynomials with distinct initial terms. We start with $F^2_{x_0}$ and perform Gaussian reduction. Denote the new set of polynomials $F^{2,norm}_{x_0}$. The general procedure is as follows.

NORMALIZE($F^i_{x_0}$)

1. The polynomials of $F^i_{x_0}$ are put into the rows of a Macaulay matrix. Perform *SplitVariable*($F^i_{x_0}, x_0$) to get new set $F^i_{x_0}$ where the x_0-part is in row-reduced echelon form. If there are polynomials without x_0-terms, put these into $F^i_{\overline{x_0}}$.

2. For each f^i in $F^i_{x_0}$ normalize it with respect to $\cup_{j=2}^{i-1} F^{j,norm}_{x_0}$.

3. Again perform *SplitVariable*($F^i_{x_0}, x_0$) to get the x_0-part in row-reduced echelon form. The polynomials containing x_0 form the set $F^{i,norm}_{x_0}$. If any polynomials do not contain x_0, then add these polynomialst to the set $F^i_{\overline{x_0}}$.

3.2. Resultants

The second tool for elimination we use is resultants, which eliminates x_0 from a pair of polynomials. Let $f_1 = a_1x_0 + b_1$ and $f_2 = a_2x_0 + b_2$ be two polynomials in $\mathbb{B}[x_0, \ldots, x_n]$. The variable x_0 has been factored out so the polynomials a_i and b_i are in $\mathbb{B}[x_1, \ldots, x_n]$. In order to find the resultant, form the 2×2 Sylvester matrix of f_1 and f_2 with respect to x_0

$$\text{Syl}(f_1, f_2, x_0) = \begin{pmatrix} a_1 & a_2 \\ b_1 & b_2 \end{pmatrix}$$

The resultant of f_1 and f_2 with respect to x_0 is then simply the determinant of this matrix, and hence a polynomial in $\mathbb{B}[x_1, \ldots, x_n]$:

$$\text{Res}(f_1, f_2, x_0) = \det(\text{Syl}(f_1, f_2, x_0)) = a_1b_2 + a_2b_1$$

We note that $\text{Res}(f_1, f_2, x_0) = a_2f_1 + a_1f_2$, too, which means that the resultant is indeed in the ideal generated by f_1 and f_2. Moreover, $\text{Res}(f_1, f_2, x_0)$ is in the elimination ideal I_1.

When both f_1 and f_2 are quadratic, then the a_i's are linear so the degree of the resultant $\text{Res}(f_1, f_2, x_0)$ is ≤ 3. More generally, since we are restricting the degree to at most d we can not take the resultants between all pairs of polynomials from F. Say $f_1 \in F^i$ and $f_2 \in F^j$. Then we have $\deg(a_1) \leq i - 1$, $\deg(a_2) \leq j - 1$, $\deg(b_1) \leq i$ and $\deg(b_2) \leq j$. Hence it follows that

$\deg(\mathrm{Res}(f,g,x_0)) \leq i+j-1$, which is certain to respect the degree bound only when $i+j \leq d+1$. Hence we can take resultants of all polynomial pairs from the sets F^i, F^j as long as $i+j \leq d+1$:

$$\mathrm{Res}_1(F^i, F^j) = \{\mathrm{Res}(f,g,x_0) | f \in F^i, g \in F^j, i+j \leq d+1\}.$$

For a set F of polynomials of differing degree, we split F into $F^d, \ldots, F^2$ and denote

$$Res_1(F) = \cup_{i+j \leq d+1} Res_1(F^i, F^j).$$

It is easy to see that $Res_1(F)$ is contained in the elimination ideal $I(F) \cap \mathbb{B}[x_1, \ldots, x_n]$, but this inclusion is in general strict.

Relation to Gröbner bases: Computing a resultant may be realised in terms of Gröbner bases, by running a slightly modified Buchberger's algorithm. We show this in the case of quadratic polynomials for ease of exposition, the general case follows the same reasoning. Assume we have a lexicographic order on the monomials, where $x_0 > x_1 > x_2$ and all other variables are smaller than x_2. Let

$$\begin{array}{rcl} f_1 & = & (x_1 + a_1)x_0 + b_1 \\ f_2 & = & (x_2 + a_2)x_0 + b_2 \end{array},$$

where a_1 and a_2 are linear combinations where neither x_1 nor x_2 appears, and b_1, b_2 do not depend on x_0. The first step of Buchberger's algorithm is to compute the S-polynomial of f_1 and f_2, and then reduce it modulo the polynomials already in the basis. Here we only consider reduction modulo the sub-basis consisting of (f_1, f_2). The leading monomial of f_1 is x_0x_1 and the leading monomial of f_2 is x_0x_2. The S-polynomial is then

$$S = x_2 f_1 + x_1 f_2 = x_0x_1a_2 + x_0x_2a_1 + x_1b_2 + x_2b_1.$$

The next step is to divide S on the two polynomials (f_1, f_2) to find the remainder that should be added to the sub-basis. The highest monomials in S are $x_0x_1x_k$, for the various x_k appearing in a_2. All of these monomials are divisible by the leading monomial of f_1. After a number of steps in the division we arrive at

$$S = a_2 f_1 + x_0x_2a_1 + x_0a_1a_2 + x_1b_2 + (x_2 + a_2)b_1.$$

The remainder now has $x_0x_2x_k$ as the highest monomials, for the different x_k appearing in a_1. All of these terms are divisible by the leading monomial of f_2. When continuing the division algorithm the $x_0a_1a_2$-terms will appear twice and cancel, so we arrive at

$$S = a_2 f_1 + a_1 f_2 + (x_1 + a_1)b_2 + (x_2 + a_2)b_1.$$

We see that the remainder (the last two terms) when reducing the S-polynomial only modulo the sub-basis (f_1, f_2) is exactly the resultant.

In a full implementation of an algorithm computing Gröbner bases the resultant would be reduced against all other polynomials in the basis. We are only interested in eliminating the terms containing x_0, so computing resultants is more efficient and straight to the point for our purpose, than computing S-polynomials followed by a full reduction.

3.3. Coefficient constraints

The next object we introduce is what is exactly required in order to close the gap that the resultants leave in the elimination ideal.

Definition 7. *Let $I(F) = (f_1, \ldots, f_m) \subseteq \mathbb{B}[x_0, \ldots, x_n]$, and write each f_i as $f_i = a_ix_0 + b_i$, where neither a_i nor b_i depends on x_0. We define the* coefficient constraint *ideal as*

$$Co_1(F) = (b_1(a_1 + 1), b_2(a_2 + 1), \ldots, b_m(a_m + 1)).$$

We note that the degrees of the generators of $\mathrm{Co}_1(F)$ have the same degrees as the generators of $Res_1(F^i, F^i)$, and it follows that we can form the coefficient constraints for F^i as long as $2i \leq d+1 \iff i \leq (d+1)/2$. In the case when $I(F)$ consists of quadratic polynomials, the generators of $\mathrm{Co}_1(F)$ will be polynomials of degree ≤ 3.

Similarly to resultants, the coefficient constraints can also be realized through familiar Gröbner basis computations when we make use of the field polynomials $x_i^2 + x_i$. We show this now for a quadratic polynomial. Consider when we have the lexicographic order where $x_0 > x_1$ and x_1 is bigger than all other variables. Let

$$f = (x_1 + a)x_0 + b,$$

where all terms in a are smaller than x_1. Compute the S-polynomial of f and $x_1^2 + x_1$:

$$S = x_1 f + x_0(x_1^2 + x_1) = x_0 x_1 a + x_0 x_1 + x_1 b.$$

The highest monomials in S are $x_0 x_1 x_j$ for the different x_j appearing in a, all of which are divisible by the leading monomial in f. We also have the quadratic monomial $x_0 x_1$ in S. Dividing S by f then gives

$$S = (a+1)f + x_0(a^2 + a) + (x_1 + a + 1)b.$$

All cross terms $x_j x_k$ in the squared linear combination a^2 will cancel since our base field has characteristic 2. Hence $x_0(a^2+a)$ will be a sum of terms $x_0 x_j^2 + x_0 x_j$ where the highest monomials are the $x_0 x_j^2$ given by the elimination order. These are all divisible by the leading monomials of the field polynomials $x_j^2 + x_j$. Continuing the division algorithm using the field equations will remove all terms in $x_0(a^2 + a)$. Assuming a starts with $x_j + x_k + \ldots$ we get

$$S = (a+1)f + x_0(x_j^2 + x_j) + x_0(x_k^2 + x_k) + \ldots + (x_1 + a + 1)b.$$

This gives a remainder $(x_1 + a + 1)b$ that does not depend on x_0, and it is exactly the coefficient constraint. In the same way as with resultants, coefficient constraints allow us to jump straight to the answer of reducing an S-polynomial modulo a given basis, instead of going through all the steps of the division in a Gröbner basis algorithm.

An important fact is that the zero set of $Co_1(F)$ lies in the projection of the zero set of $I(F)$ onto $\mathbb{F}_2^n$.

Lemma 8. *$Z(Co_1(F)) \supseteq \pi_1(Z(I(F)))$.*

Proof. A point $\mathbf{p} \in \mathbb{F}_2^n$ is not in the zero set $Z(\mathrm{Co}_1(F))$ only when for some i we have $a_i(\mathbf{p}) = 0$ and $b_i(\mathbf{p}) = 1$. But then for both the two liftings of $\mathbf{p}$ to $\mathbb{F}_2^{n+1}$: $\mathbf{p}_0 = (0, \mathbf{p})$ and $\mathbf{p}_1 = (1, \mathbf{p})$ we have $f_i(\mathbf{p}_j) = 1$. Therefore $\mathbf{p} \notin \pi_1(Z(I(F)))$, and so we must have $Z(\mathrm{Co}_1(F)) \supseteq \pi_1(Z(I(F)))$. ∎

By the Lemmas 5 and 8, it follows that the coefficient constraint ideal lie in the elimination ideal. We can now use this ideal to describe the full elimination ideal, which turns out to be generated exactly by $\mathrm{Res}_1(F)$ and $\mathrm{Co}_1(F)$.

Theorem 9. *Let $I(F) = (f_1, \ldots, f_m) \subseteq \mathbb{B}[x_0, \ldots, x_n]$ be an ideal generated by a set F of Boolean polynomials. Then*

$$I(F) \cap \mathbb{B}[x_1, \ldots, x_n] = I(\mathrm{Res}_1(F), Co_1(F)).$$

Proof. By Lemma 5 we have

$$\pi_1(Z(I(F))) = Z(I(F) \cap \mathbb{B}[x_1, \ldots, x_n]).$$

We know that

$$I(F) \cap \mathbb{B}[x_1, \ldots, x_n] \supseteq I(\mathrm{Res}_1(F), \mathrm{Co}_1(F)),$$

which implies that

$$\pi_1(Z(I(F))) = Z(I(F) \cap \mathbb{B}[x_1, \ldots, x_n]) \subseteq Z(\mathrm{Res}_1(F)) \cap Z(\mathrm{Co}_1(F)).$$

Conversely, let a point $\mathbf{p} \in Z(\mathrm{Res}_1(F)) \cap Z(\mathrm{Co}_1(F))$ be given. Then $\mathbf{p}$ has two liftings to points in $\mathbb{F}_2^{n+1}$: $\mathbf{p}_0 = (0, \mathbf{p})$ and $\mathbf{p}_1 = (1, \mathbf{p})$. We will show that at least one of $\mathbf{p}_0$ or $\mathbf{p}_1$ is contained in $Z(I(F))$. Let $f_i = x_0 a_i + b_i$ be an element in F. Since $\mathbf{p}$ vanishes on $\mathrm{Co}_1(F)$, the following are the possible values for the terms in f_i.

$a_i(\mathbf{p})$	$b_i(\mathbf{p})$
0	0
1	0
1	1

Note that $\mathrm{Co}_1(F)$ excludes $(a_i(\mathbf{p}), b_i(\mathbf{p}))$ from taking the value $(0, 1)$. Since $\mathbf{p}$ also vanishes on the resultant ideal, there cannot be two f_i and f_j such that $(a_i(\mathbf{p}), b_i(\mathbf{p}))$ takes the value $(1, 0)$ and $(a_j(\mathbf{p}), b_j(\mathbf{p}))$ takes the value $(1, 1)$, since in that case the resultant $a_i b_j + a_j b_i$ would not vanish. This means that either 1) : $\{(a_i(\mathbf{p}), b_i(\mathbf{p})) | 1 \leq i \leq m\} \subseteq \{(0, 0), (1, 0)\}$, or 2) : $\{(a_i(\mathbf{p}), b_i(\mathbf{p})) | 1 \leq i \leq m\} \subseteq \{(0, 0), (1, 1)\}$. In case 1), the lifting $\mathbf{p}_0$ is in the zero set $Z(I(F))$. In case 2) the lifting $\mathbf{p}_1$ is in the zero set of $Z(I(F))$. This shows that $Z(\mathrm{Res}_1(F)) \cap Z(\mathrm{Co}_1(F))$ lifts to $Z(I(F))$, which means that

$$\pi_1(Z(I(F))) = Z(I(F) \cap \mathbb{B}[x_1, \ldots, x_n]) \supseteq Z(\mathrm{Res}_1(F)) \cap Z(\mathrm{Co}_1(F))$$

as desired. ■

In general, for an ideal $I(F) \subseteq \mathbb{B}[x_0, \ldots, x_n]$, this process can be iterated eliminating more variables from $I(F)$. We denote by $\mathrm{Res}_k(F)$ and $Co_k(F)$ the iterative application of the resultant and the coefficient constraint ideal with respect to a sequence $x_0, \ldots, x_{k-1}$ of variables to be eliminated. Note that both Lemma 8 and Proposition 9 easily generalize to this case. We can also generalize Theorem 9 as follows.

Corollary 10. *For $I(F) = (f_1, \ldots, f_m)$ in $\mathbb{B}[x_0, \ldots, x_n]$, then*

$$I(F) \cap \mathbb{B}[x_k, \ldots, x_n] = I(\mathrm{Res}_k(F), Co_k(F)).$$

This enables us to actually compute the elimination ideal *independent of monomial order*, in contrast to approaches using Gröbner bases (see [2, 3]). Moreover, one could find the elimination ideal by successively eliminating $x_0, \ldots, x_{k-1}$ using Corollary 10 by the following algorithm:

0. $F_0 = F$,
1. $F_1 =$ generators of $\mathrm{Res}_1(F_0) + Co_1(F_0)$,
2. $F_2 =$ generators of $\mathrm{Res}_2(F_1) + Co_2(F_1)$,

   ...

i. $F_i =$ generators of $\mathrm{Res}_i(F_{i-1}) + Co_i(F_{i-1})$,

   ...

We show that this simple solving algorithm already gives a complexity that is better than the trivial $\mathcal{O}(2^n)$.

Lemma 11. *If the degree of the polynomials in F_0 is 2, then the maximal degree of the polynomials in F_i is at most $2^i + 1$.*

Proof. We show this by induction. The statement is clearly true for $i = 0$. Assume the statement is true for $i = j$, so the maximal degree of the polynomials in F_j is $2^j + 1$. The generators for both $\mathrm{Res}_{j+1}(F_j)$ and $Co_{j+1}(F_j)$ are computed by multiplying two polynomials, where the first has degree at most 2^j (because x_j has been factored out), and the other has degree at most $2^j + 1$. The generators of $Res_{j+1}(F_j)$ and $Co_{j+1}(F_j)$ then get maximal degree $2 \cdot 2^j + 1 = 2^{j+1} + 1$, as desired. ■

Theorem 12. *The maximal degree of any polynomial in any F_i when running the elimination algorithm above is upper bounded by $n - \log_2(n) + 1$ and F_0 only has quadratic polynomials.*

Proof. Let D be the maximal degree encountered during the elimination algorithm. From Lemma 11 we know that $D = max_i(min\{2^i + 1, n - (i - 1)\})$. We have $2^i + 1 < n - i + 1$ for small i, but the function $2^i + 1$ is increasing and the function $n - i + 1$ is decreasing when i increases, so at some point the inequality stops being true. Let i_0 be the crossover point, that is, let i_0 be defined as the integer such that both $2^{i_0} + 1 \leq n - i_0 + 1$ and $2^{i_0+1} + 1 > n - i_0$ holds. Then $D = \max\{2^{i_0} + 1, n - i_0\}$. It is easy to see that $2^{log_2(n)-1} + 1 = n/2 + 1 \leq n - log_2(n) + 2$ and $2^{log_2(n)} + 1 = n + 1 > n - log_2(n)$ when $n \geq 1$, so $i_0 = log_2(n) - 1$. The theorem then follows since $D = \max\{2^{log_2(n)-1} + 1, n - log_2(n) + 1\} = n - log_2(n) + 1$. ■

In [13, p. 315] it is stated that the maximal degree of polynomials occurring in the computation of Gröbner bases over $\mathbb{B}[x_0, \ldots, x_n]$ is $n + 1$. Theorem 12 improves on this, in the sense that a solving algorithm using elimination of variables only needs to consider polynomials of degree up to $n - log_2(n) + 1$.

Applying the straight-forward elimination of variables in practice leads to problems however, due to the initial exponential growth of degrees from the resultants and coefficient constraints with each elimination. With many variables the number of monomials quickly becomes too large for a computer to work with, which is why we suggest in this paper to only compute polynomials of degree $\leq d$, where d is a free parameter. Note that Corollary 10 is only valid if we do not restrict the maximum degree allowed.

3.4. Syzygies

The last concept we will define is that of *syzygies*. In fact, syzygies will give a more general view on resultants and coefficient constraints. The reason for this extension is that for degree d, then in general both $\deg(\mathrm{Res}(F^d, F^d)) = 2d - 1$, and $\deg(\mathrm{Co}(F^d)) = 2d - 1$. This means that in order to compute the generators for $I(\cup_{i=2}^{d} F^i) \cap \mathbb{B}[x_1, \ldots, x_n]$, we would need to allow polynomials of degree $2d - 1$. When restricting the maximum allowed degree to d, this implies that resultants and coefficient constraints cannot compute the full elimination ideal. Syzygies fills this gap.

Definition 13. *Let $a_1, \ldots, a_\ell$ be boolean polynomials in $\mathbb{B} = \mathbb{B}[x_1, \ldots, x_n]$. Let*

$$\mathbb{B}^\ell = \mathbb{B}\varepsilon_1 \oplus \cdots \oplus \mathbb{B}\varepsilon_\ell$$

be the free $\mathbb{B}$-module of rank ℓ, where $\varepsilon_1, \ldots, \varepsilon_\ell$ is an (abstract) basis. The syzygy module S for polynomials a_i is the submodule of $\mathbb{B}^\ell$ consisting of all $r_1\varepsilon_1 + \cdots + r_\ell\varepsilon_\ell \in \mathbb{B}^\ell$ which form a relation

$$r_1 a_1 + \cdots + r_\ell a_\ell = 0.$$

We may also specify natural numbers $d_1, \ldots d_\ell$ such that $\deg a_i \leq d_i$. We let the module $\mathbb{B}^\ell$ now be graded by letting ε_i have degree d_i. Then the syzygies of degree $\leq d$, $S^{\leq d}$ consists of the syzygies $\sum_i r_i\varepsilon_i$ such that $\deg(r_i) + d_i \leq d$.

Syzygies are connected to Gröbner bases in the literature in relation to optimizations of Buchberger's algorithm. In fact, most known approaches to Gröbner bases (as for example [10, 11]), reduce to computing the module of syzygies over the polynomial ring $K[x_0, \ldots, x_n]$ for some field K.

In [12, Ch.3] it is shown that whenever we get a reduction to 0 for an S-polynomial in Buchberger's algorithm, this reduction corresponds to a syzygy. One way to find the syzygies of some polynomials $(f_1, \ldots, f_m)$ is therefore to save the ones encountered when reducing S-polynomials in Buchberger's algorithm for computing Gröbner bases. In our work we also consider syzygies, but we generate a basis for them directly, without going through reductions of S-polynomials.

The approach in this paper is not to compute Gröbner bases, but rather find ways to eliminate variables from the system of polynomials. Note that it is known that one can compute the elimination ideal by using Gröbner bases ([2, 3]). Our approach is related but differs from the Gröbner bases approaches considered in the literature.

The idea in this paper is to find more efficient ways of computing the vector space

$$\langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \cdots \cup L^{d-2} F^2 \rangle \cap \mathbb{B}[x_1, \ldots, x_n].$$

Let us indicate how syzygies enter here. Given a set of polynomials

$$F : x_0 a_1 + b_1, x_0 a_2 + b_2, \ldots, x_0 a_m + b_m,$$

let V be the linear space of all expressions

$$l_1(x_0 a_1 + b_1) + \cdots + l_m(x_0 a_m + b_m)$$

with certain degree restrictions $\deg(l_i) \leq d_i$ or equivalently $l_i \in L^{d_i}$. We want to eliminate x_0 and compute the space $V \cap \mathbb{B}[x_1, \ldots, x_n]$. By a suitable reduction process we shall obtain a set of polynomials

$$F' : x_0 a'_1 + b'_1, x_0 a'_2 + b'_2, \ldots, x_0 a'_k + b'_k.$$

Let V' be the linear space of all expressions

$$l'_1(x_0 a'_1 + b'_1) + \cdots + l'_l(x_0 a'_k + b'_k) \tag{2}$$

where l'_i now is a polynomial in $L^{e_i}_{\frac{}{x_0}}$ (for appropriate e_i) where $L^{e_i}_{\frac{}{x_0}}$ is L^{e_i} with all x_0-terms omitted. The essential step in computing the elimination space $V \cap \mathbb{B}[x_1, \ldots, x_n]$ turns out to be to compute $V' \cap \mathbb{B}[x_1, \ldots, x_n]$. But we see that if an element (2) is in the latter intersection, we must have

$$l'_1 a'_1 + \cdots + l'_k a'_k = 0,$$

so $l'_1 \varepsilon_1 + \cdots + l'_k \varepsilon_k$ is a syzygy for the a'_i. For each such syzygy we get by (2) a corresponding element in the intersection $V' \cap \mathbb{B}[x_1, \ldots, x_n]$:

$$l'_1 b'_1 + \cdots + l'_k b'_k$$

and this intersection is the space of all such elements as we vary over the syzygies of the a'_i.

The following illustrate how the approach in this paper differs from Gröbner bases.

1. In this paper we compute the syzygies directly in the Boolean ring $\mathbb{B}[x_1, \ldots, x_n]$, which means that the field equations are encoded into our computations in contrary to Gröbner bases which are computed over the polynomial ring $\mathbb{F}_2[x_1, \ldots, x_n]$ and thus needing the field equations to be added to the systems of equations.

2. In the approach of this paper we only need to compute syzygies on the a_j^{i-1} terms, and these *have degree one less* than the f^i's.

3. The approach here avoids the chain of reductions as done in Buchberger's algorithm since syzygies are computed directly. The trivial syzygies here are Koszul syzygies $a_l^{j-1}\varepsilon_k^{i-1} - a_k^{i-1}\varepsilon_l^{j-1}$, and Boolean syzygies$(a_k^{i-1}+1)\varepsilon_k^{i-1}$ and any multiples of these allowed by the degree restriction d.

4. The approach here is "straight to the point" meaning that we eliminate variables in a straight forward way avoiding precise term orderings apart from the elimination order $x_0 > x_1 > \ldots$.

The Boolean syzygies are special because they only occur for characteristic 2. In the following we show that these two types of trivial syzygies give the resultants and coefficient constraints when applied to the f^i's.

3.5. Syzygies between linear polynomials

Let $a_1, \cdots a_\ell$ be Boolean polynimals in $\mathbb{B}[x_1, \ldots, x_n]$ of degree ≤ 1. Recall that $L_{\overline{x_0}} = \{1, x_1, \ldots, x_n\}$ so this is the same as saying that $a_i \in \langle L_{\overline{x_0}} \rangle$. After suitable Gaussian elimination we may assume they are ordered such that the intial terms (in this case a variable or simply the constant 1)

$$\mathrm{in}(a_1) > \mathrm{in}(a_2) > \cdots > \mathrm{in}(a_\ell). \tag{3}$$

Proposition 14. *Let $S \subseteq \mathbb{B}_{\varepsilon_1} \oplus \cdots \oplus \mathbb{B}_{\varepsilon_\ell}$ be the syzygy module for $a_1, \ldots, a_\ell$, where each ε_i has degree 1. There are the following syzygies in S:*

1. *Koszul syzygies $a_j\varepsilon_i + a_i\varepsilon_j$,*
2. *Boolean syzygies $(a_i+1)\varepsilon_i$.*

Let K (resp. B) be the linear spaces generated by the Koszul (resp. Boolean) syzygies, and let $m \geq 2$. Then $S^{\leq m}$ is $\langle L_{\overline{x_0}}^{m-2} \rangle K + \langle L_{\overline{x_0}}^{m-2} \rangle B$.

Proof. Let $\mathbf{r} = \sum_i r_i\varepsilon_i$ be a syzygy of degree $\leq m$ so each $\deg(r_i) \leq m-1$. Suppose a term in r_i is $n \cdot \mathrm{in}(a_j)$ where $i \leq j$ and $\deg(n)+1 \leq m-1$ (so $n \in \langle L_{\overline{x_0}}^{m-2} \rangle$).If $i < j$ we subtract $n \cdot (a_i\varepsilon_j + a_j\varepsilon_i)$ from $\mathbf{r}$, and if $i = j$ we subtract $(a_i+1)\varepsilon_i$. Continuing this way we get a syzygy $\mathbf{r}' = \sum_i r_i'\varepsilon_i$ such that r_i' contains no term $\mathrm{in}(a_j)$ where $j \leq i$. We show that $\mathbf{r}' = 0$. This will prove the proposition. So we have the relation $\sum_i r_i'a_i = 0$. Let $\mathrm{in}(a_1) = x_1$, say. Then no a_j for $j \geq 2$ contains the variable x_1 by the assumption (3). But also no r_j' contains the variable x_1 by construction. Hence the only terms in the relation above that contains x_1 are the terms in $r_1'x_1$. Hence we must have $r_1' = 0$. In this manner we may continue and we get that all $r_i' = 0$. ∎

When considering a system of independent quadratic polynomials $F^2 = \{f_1, \ldots, f_m\}$ where each $f_i = x_0a_i + b_i$ or $f_i = b_i$ (if f does not contain x_0), it follows that the a_i's are polynomials of degree ≤ 1. By Proposition 14 we know that the syzygy module is generated by Koszul and Boolean syzygies, which implies that if we map ε_i to b_i and ε_j to b_j in the Koszul syzygies we generate exactly the resultant $\mathrm{Res}(f_i, f_j; x_0) = a_jb_i + a_ib_j$. Similarly, mapping ε_i to b_i in the Boolean syzygies, we generate exactly the Coefficient constraint $\mathrm{Co}(f_i; x_0) = (a_i+1)b_i$. This implies the following straight forward additional result.

Corollary 15. *Let $x_0a_i + b_i$ for $i = 1, \ldots, m$ be quadratic polynomials in $\mathbb{B}[x_0, \ldots, x_n]$. The linear span of the resultants and coefficient constraints in $\mathbb{B}[x_1, \ldots, x_n]$ is precisely the image of the composition*

$$K + B \subseteq (\mathbb{B}^m)^2 \xrightarrow{\phi} \langle L_{\overline{x_0}}^3 \rangle,$$

where ϕ sends $\varepsilon_i \mapsto b_i$.

Note how this relates to compute the intersection $\langle L_{\overline{x_0}}^1 F_{x_0}^2 \rangle \cap \mathbb{B}[x_1, \ldots, x_n]$ where now the l_i's do not contain x_0. By Proposition 14 the above is equivalent to find all solutions $(l_1, \ldots, l_m)$, where $l_i \in \langle L_{\overline{x_0}}^1 \rangle$ that satisfy $l_1a_1 + l_2a_2 + \cdots + l_{r_2}a_{r_2} = 0$. This is the linear span of the Koszul and Boolean syzygies. Corollary 15 above gives that the intersection is generated by the resultants and the coefficient constraints.

3.6. Syzygies between polynomials of degrees ≥ 2

When the degree of some a_i's are greater than 1 there may be other syzygies that are not generated by the Koszul and Boolean syzygies. Let $a_1^1, \dots a_{\ell_1}^1$ be polynomials of degree ≤ 1. By suitable Gaussian elimination we may assume the initial terms are such that:

$$\text{in}(a_1^1) > \text{in}(a_2^1) > \cdots > \text{in}(a_{\ell_1}^1). \tag{4}$$

Let $a_1^d, \dots, a_{\ell_d}^d$ be polynomials of degree $\leq d$. We perform reduction operations as follows: If a term of a_i^d is of the form $t \cdot \text{in}(a_j^1)$ where t is a monomial of degree $\leq d-1$, we replace a_i^d by $a_i^d - t \cdot a_j^1$. We then eventually get:

$$\text{No term of } a_i^d \text{ is } t \cdot \text{in}(a_j^1) \text{ where } t \text{ is a monomial of degree} \leq d-1. \tag{5}$$

Secondly we may perform Gaussian elimination on the a_i^d such that:

$$\text{in}(a_1^d) > \text{in}(a_2^d) > \cdots > \text{in}(a_{\ell_d}^d). \tag{6}$$

Suppose we have given a_i^d as above for each $1 \leq d \leq D$ and $i = 1, \dots, \ell_d$. Let

$$\mathbb{B}^L = \mathbb{B}^{\ell_1} \oplus \cdots \oplus \mathbb{B}^{\ell_D}$$

where $\mathbb{B}^{\ell_d} = \mathbb{B}\varepsilon_1^d \oplus \cdots \oplus \mathbb{B}\varepsilon_{\ell_d}^d$ and we set ε_j^d to have degree d. There is a map

$$\mathbb{B}^L \to \mathbb{B}, \quad \varepsilon_i^d \mapsto a_i^d$$

and the syzygy module $S \subseteq \mathbb{B}^L$ is the kernel of this map.

Suppose now we have a total order on the terms of $\mathbb{B}$. We make a term order on $\mathbb{B}^L$ by letting terms $s\varepsilon_j^e < t\varepsilon_i^d$ if:

- $e < d$, or
- $e = d$ and $j < i$, or
- $e = d, j = i$ and $s < t$

Theorem 16. *Given polynomials a_i^d of degree $\leq d$, for each $1 \leq d \leq D$ and suppose for each d they fulfill Condition* (6) *above. There are the following syzygies:*

1. *Koszul syzygies $a_j^d\varepsilon_k^e + a_k^e\varepsilon_j^d$ where $e < d$ or $e = d$ and $k < j$. For given sum $d+e$ denote by K^{d+e} the linear space these syzygies generate.*
2. *Boolean syzygies $(a_j^d+1)\varepsilon_j^d$. For given d denote by B^{2d} the linear space these syzygies generate.*
3. *For each $m \geq 2$ syzygies*
$$\mathbf{r} = \sum_{\substack{d=1,\dots,m \\ i=1,\dots,\ell_d}} r_i^{m-d}\varepsilon_i^d$$
where r_i^{m-d} has degree $\leq m-d$ and no term of $\mathbf{r}$ is $n \cdot t$ where t is the initial term of a syzygy in K^e or B^e and $\deg(n) + e \leq m$.

For a given m in 3. denote by $R^{\leq m}$ the linear space of such syzygies.

a. Then for $m \geq 2$ we have:

$$S^{\leq m} = \sum_{d=2}^{m} S^{\leq m-d}K^d + \sum_{d=2}^{m} S^{\leq m-d}B^d + R^{\leq m}. \tag{7}$$

b. Suppose in addition the a_i^d fulfill the Condition (5) *above. Then we may let $R^{\leq m}$ be the space of all syzygies of type 3. where the coefficient r_i^{m-1} of the a_i^1 vanish, and we still have the above identity* (7)*.*

Proof. Given a syzygy of degree $\leq m$

$$\mathbf{s} = \sum_{\substack{d=1,\dots,m \\ i=1,\dots,\ell_d}} s_i^{m-d}\varepsilon_i^d.$$

If a term in $\mathbf{s}$ is a product $n \cdot t$ where t is the initial term of a syzygy $\mathbf{s}'$ in K^p or B^p with $\deg(n) + p \leq m$, we replace $\mathbf{s}$ by $\mathbf{s} - n \cdot \mathbf{s}'$. In this way we continue and in the end we get syzygy as in 3. This proves the identity (7) above.

Suppose now the Condition (5) is also fulfilled. Let the following relation be of Type 3. :

$$\sum_{i=1}^{\ell_1} r_i^{m-1} a_i^1 + \sum_{\substack{d=2,\dots,m \\ i=1,\dots,\ell_d}} r_i^{m-d}\varepsilon_i^d.$$

Let $x_1 = \mathrm{in}(a_1^1)$. Then no term of any other a_i^d contains x_1 and also no r_i^{m-d} contains x_1. But then the relation above is only possible if $r_1^1 = 0$. In this way we may continue and get all $r_i^1 = 0$ except possibly if $\mathrm{in}(a_j^1)$ is the constant 1 (in which case we must have i the last index ℓ_1). But then by the reduction process using $a_{\ell_1}^1$, none of the a_i^d for $d \geq 2$ contains a term of degree $< d$ and similarly no term of the r_j^{m-d} contains a term of degre $< m - d$. But then in the relation

$$r_{\ell_1}^{m-1} \cdot 1 + \sum_{\substack{d=2,\dots,m \\ i=1,\dots,\ell_d}} r_i^{m-d} a_i^d,$$

the left side has degree $\leq m-1$ while the right side has all terms of degree m. Hence $r_{\ell_1}^{m-1} = 0$. ∎

We now present the algorithm to compute $R^{\leq m}$ under the assumption of Conditions (6) and (5).

Algorithm to compute $R^{\leq m}$

1. Set $KB_{in}^{\leq 1}, R_{in}^{\leq 1}$ equal to 0. Let $m := 2$.
2. Let KB_{in}^m consist of all pairs (t, m) where t is the initial term of a Koszul syzygy in K^m or a Boolean syzygy in B^m.
3. $KB_{in}^{\leq m} = KB_{in}^{\leq m-1} \cup KB_{in}^m$.
4. If $m = 2$ let $R^2 = 0$. If $m \geq 3$ then R^m consist of all syzygies

$$\mathbf{r} = \sum_{\substack{d=2,\dots,m \\ i=1,\dots,\ell_d}} r_i^{m-d}\varepsilon_i^d$$

   where r_i^{m-d} has degree $\leq m-d$ and no term of $\mathbf{r}$ is a product of monomials $n \cdot t$ where:
   - $(t, p) \in R_{in}^{\leq m-1} \cup KB_{in}^{\leq m-1}$ and $\deg(n) + p \leq m$.
   - $n = 1$ and $(t, m) \in KB_{in}^m$
5. Perform Gaussian elimination on R^m and let R_{in}^m consists of all pairs (t, m) where t is an initial term of a syzygy in R^m.
6. $R_{in}^{\leq m} = R_{in}^{\leq m-1} \cup R_{in}^m$.

7. If m is less than the stop bound then $m := m + 1$ and go to 2.

As for the actual compuation of the syzygies in Step 4, this can be done by taking the r_i^{m-d} to be linear combinations of the allowed terms (with unknown coefficients), and then solving a system of linear equations.

Proposition 17. *With the algorithm above, then*

$$R^{\leq m} = \sum_{d \geq 3} S^{\leq m-d} R^d.$$

Proof. This is clear by construction. ∎

Our applications of Theorem 16 are typically for $D = 1$ or 2. (This occurs for sets $F^2, \ldots, F^d$ where $d = 3$ or 4.) We are then interested in the syzygies $S^{\leq 2}$ and $S^{\leq 3}$. These are given as follows:

$$\begin{aligned} S^{\leq 2} &= K_2 + B_2 \\ S^{\leq 3} &= \langle L^1 \rangle K_2 + K_3 + \langle L^1 \rangle B_2 + R^3. \end{aligned}$$

4. Elimination of variables from systems of Boolean equations

Previous work on solving Boolean equation systems in ANF form include the XL and XSL algorithms [6, 7]. In those approaches one multiplies all equations with all monomials up to some fixed degree. If the degree is big enough, one could hope to have more equations than monomials in the system, and hence solve the system by re-linearization.

One can also use the approach of multiplying all polynomials with all monomials up to some degree to eliminate the variable x_0. Indeed, after generating the Macaulay matrix of the full set of polynomials one can perform Gaussian elimination on terms containing x_0. If we have more independent polynomials than x_0-terms we are certain to end up with some equations with no terms depending on x_0. This procedure can obviously be iterated to eliminate a sequence of variables $x_0, x_1, \ldots$ just by using Gaussian elimination on the set of polynomials after multiplying with all allowed monomials.

When we have polynomials of low degree in F, the complexity of multiplying all of them with all monomials of allowed degree quickly becomes very large when d increases. We aim to eliminate x_0 for the sets $F^d, F^{d-1}, \ldots, F^2$ in a more efficient way motivated by the tools developed in the previous section.

4.1. Bounding the degree to $d \leq 3$

The lowest degree possible to make meaningful use of the elimination techniques is to set the degree bound at $d = 3$. Here the input polynomials are cubic and quadratic, which means that we consider the sets F^3 and F^2. In the following procedure these sets will be modified to only include polynomials respecting the degree constraint $d \leq 3$ while eliminating the variable x_0.

Elimination Procedure

1. We start by splitting the set F^2 into subsets $F_{x_0}^{2,norm}$ and $F_{\overline{x_0}}^2$ using $SplitVariable(F^2, x_0)$. We increase F^3, by adding $x_0 F_{\overline{x_0}}^2$ and $(x_0 + 1)F_{x_0}^{2,norm}$ to F^3. Note that we only multiply the sets $F_{x_0}^{2,norm}$ and $F_{\overline{x_0}}^2$ with $(x_0 + 1)$, resp. x_0 and not with all linear polynomials.

2. With the new sets, we normalize F^3 with $F_{x_0}^{2,norm}$ as basis, producing $F_{x_0}^{3,norm}$ and $F_{\overline{x_0}}^3$. Note that we only normalize w.r.t. $F_{x_0}^{2,norm}$. It is quite probable that there are monomials $x_0 n$ in polynomials in F^3 where n is an initial term of some polynomial in $F_{\overline{x_0}}^2$. However we do not perform any operation in this case. Only if the term $x_0 n$ is $x_i \cdot x_0 n'$ where $x_0 n'$

is an inital term of a polynomial in $F^{2,norm}_{x_0}$, do we perform a reduction. Since $F^{2,norm}_{x_0}$ is usually considerably smaller than $F^2_{\overline{x_0}}$, this saves many operations. The reason that we may save this work is that we add $x_0F^2_{\overline{x_0}}$ to F^3 in Step 1. above. The Gaussian elimimination in $SplitVariable$ allows us however to peform the reductions with these "all at once" in analogy with the F4 algorithm, [10]. The normalization process removes a lot of the degree ≤ 3 monomials depending on x_0 in $F^3_{x_0}$, and may eliminate x_0 completely from many of them. Polynomials in F^3 not depending on x_0 in the first place are simply added to $F^3_{\overline{x_0}}$.

3. The final step is to compute resultants and coefficient constraints from the set $F^{2,norm}_{x_0}$. We create $Res(F^{2,norm}_{x_0})$ and $Co(F^{2,norm}_{x_0})$ and join these sets with $F^3_{\overline{x_0}}$.

The outputs of the procedure are $F^3_{\overline{x_0}} := F^3_{\overline{x_0}} \cup Res(F^{2,norm}_{x_0}) \cup Co(F^{2,norm}_{x_0})$ and $F^2_{\overline{x_0}}$, sets of cubic and quadratic polynomials that do not depend on x_0. The following theorem shows that by following this procedure we have not lost anything essential, in the sense that all polynomials in $F^2_{\overline{x_0}}$ multiplied with $L_{\overline{x_0}} = \{x_1, \ldots, x_n\}$ together with $F^3_{\overline{x_0}}$ still generate the whole $\langle F^3 \cup LF^2\rangle \cap \mathbb{B}[x_1, \ldots, x_n]$.

Theorem 18. *Let F^3 and F^2 be the input sets of polynomials, and $F^3_{\overline{x_0}}$ and $F^2_{\overline{x_0}}$ be the outputs of the elimination procedure.*

a. The linear span

$$\langle F^3 \cup L^1F^2\rangle = \langle F^{3,norm}_{x_0} \cup L^1_{\overline{x_0}}F^{2,norm}_{x_0} \cup F^3_{\overline{x_0}} \cup L^1_{\overline{x_0}}F^2_{\overline{x_0}}\rangle.$$

b. The elimination space

$$\langle F^3 \cup L^1F^2\rangle \cap \mathbb{B}[x_1, \ldots, x_n] = \langle F^3_{\overline{x_0}} \cup L^1_{\overline{x_0}}F^2_{\overline{x_0}}\rangle.$$

Proof. a. It is clear that we have the inclusion $\supseteq$. Let us now prove that we have inclusion $\subseteq$. First note that we have the following decompositions

$$\langle F^2\rangle = \langle F^{2,norm}_{x_0} \cup F^2_{\overline{x_0}}\rangle$$

and

$$\langle L^1\rangle = \langle x_0\rangle + \langle L^1_{\overline{x_0}}\rangle, \quad \langle L^1\rangle = \langle x_0 + 1\rangle + \langle L^1_{\overline{x_0}}\rangle.$$

This gives

$$\begin{aligned} \langle L^1F^2\rangle &= \langle L^1F^{2,norm}_{x_0}\rangle + \langle L^1F^2_{\overline{x_0}}\rangle \\ &= (x_0+1)\langle F^{2,norm}_{x_0}\rangle + \langle L^1_{\overline{x_0}}F^{2,norm}_{x_0}\rangle + x_0\langle F^2_{\overline{x_0}}\rangle + \langle L^1_{\overline{x_0}}F^2_{\overline{x_0}}\rangle. \end{aligned} \tag{8}$$

Now by the normalization procedure in Part 2. above:

$$\langle F^3\rangle + (x_0+1)\langle F^{2,norm}_{x_0}\rangle + x_0\langle F^2_{\overline{x_0}}\rangle \subseteq \langle F^{3,norm}_{x_0} \cup F^3_{\overline{x_0}} \cup L^1_{\overline{x_0}}F^{2,norm}_{x_0}\rangle. \tag{9}$$

Hence putting (8) and (9) together we obtain

$$\langle F^3\rangle + \langle L^1F^2\rangle \subseteq \langle F^{3,norm}_{x_0}\rangle + \langle L^1_{\overline{x_0}}F^{2,norm}_{x_0}\rangle + \langle F^3_{\overline{x_0}}\rangle + \langle L^1_{\overline{x_0}}F^2_{\overline{x_0}}\rangle,$$

which gives part a.

b. By the identity in a. we see that

$$\begin{aligned} &\langle F^3 \cup L^1F^2\rangle \cap \mathbb{B}[x_1, \ldots, x_n] \\ = \quad &\langle F^3_{\overline{x_0}} \cup L^1_{\overline{x_0}}F^2_{\overline{x_0}}\rangle + \langle F^{3,norm}_{x_0} \cup L^1_{\overline{x_0}}F^{2,norm}_{x_0}\rangle \cap \mathbb{B}[x_1, \ldots, x_n]. \end{aligned} \tag{10}$$

Let $x_0 a_i^1 + b_i^2$ be the elements of $F_{x_0}^{2,norm}$ and $x_0 a_i^2 + b_i^3$ be the elements of $F_{x_0}^{3,norm}$. Then any element of the intersection on the right side of (10), comes from an expression

$$\sum_i l_i(x_0 a_i^1 + b_i^2) + \sum_i c_i(x_0 a_i^2 + b_i^3),$$

where the l_i are linear polynomials in $\mathbb{B}[x_1, \ldots, x_n]$ and the c_i constants. We must have the relation

$$\sum_i l_i a_i^1 + \sum_i c_i a_i^2 = 0.$$

But due to the elements of $F_{x_0}^{3,norm}$ being normalized with respect to $F_{x_0}^{2,norm}$, it follows by Theorem 16 that in any such relation we have the $c_i = 0$ and the syzygy $\sum_i l_i \varepsilon_i^1$ a linear combination of Koszul and Boolean syzygies. By Corollary 15 the last intersection in (10) is then a linear combination of resultants and coefficient contraints, and these have already been put into $F_{\overline{x_0}}^2$. This proves part b. ■

Optional addition to the algorithm.
A. At several steps we added polynomials to $F_{\overline{x_0}}^3$. It may happen that such a polynomial has degree 2. We may optionally check the degree of each polynomial that we add and in case of degree 2, we put it in $F_{\overline{x_0}}^2$ instead. This will improve the algorithm, since now

$$\langle F_{\overline{x_0}}^3 \cup L_{\overline{x_0}} F_{\overline{x_0}}^2 \rangle$$

will normally be larger than the intersection in Theorem 18b.

B. It may also happen that the normalization process produces polynomials in $F_{x_0}^{3,norm}$ of degree 2. We could test for this and if so, add this to $F_{x_0}^2$ and start the algorithm for eliminating x_0 over again. The new

$$\langle F^3 \cup LF^2 \rangle$$

will normally be larger than the original one, and so we normally get a larger elimination space in Theorem 18b. Doing this allows us to "compute with terms of degree 4 by only computing with terms of degree 3": Suppose we produce a polynomial of degree 2

$$h = \sum_i c_i f_i^3 + \sum_i l_i f_i^2$$

where the c_i are constants and the l_i are linear. Putting h in F^2, we can then multiply it with a linear polynomial l' to produce

$$h \cdot l' = \sum_i c_i l' f_i^3 + \sum_i l_i l' f_i^2,$$

and we see that the terms in the right expression will generally be of degree 4.

C. One could also systematically find all polynomials of degree 2 in $\langle F^3 \cap LF^2 \rangle$ by adding a procedure *SplitDeg*(F^3, F^2) which uses a degree order and finds a basis $F^{2\prime}$ for the degree 2 polynomials in $\langle F^3 \cup LF^2 \rangle$. If the space $\langle F^{2\prime} \rangle$ is larger than $\langle F^2 \rangle$ we could use F^3 and $F^{2\prime}$ as our new sets of polynomials. The procedure *SplitDeg* will however be considerably more computationally heavy than *SplitVariable*: For degree 3 polynomials, the latter essentially eliminates all monomials $x_0 t$ where t is of degree 2 in $x_1, \ldots, x_n$ and there are $\binom{n}{2}$ such, while *SplitDeg* would eliminate all monomials of degree 3 in $x_0, \ldots, x_n$ and there are $\binom{n}{3}$ such.

Theorem 18 generalizes easily to eliminating several variables using the elimination procedure.

Corollary 19. *Let $L^1_{\overline{x_0,\ldots,x_{k-1}}} = \{1, x_k, \ldots x_n\}$ be the subset of L not containing the variables $x_0, \ldots, x_{k-1}$. Let $F^2_{\overline{x_0,\ldots,x_{k-1}}}$ and $F^3_{\overline{x_0,\ldots,x_{k-1}}}$ be the result of applying the elimination procedure above k times to the input sets F^3, F^2, eliminating one variable at the time in the sequence $x_0, \ldots, x_{k-1}$. Then*

$$\langle F^3_{\overline{x_0,\ldots,x_{k-1}}} \cup L^1_{\overline{x_0,\ldots,x_{k-1}}} F^2_{\overline{x_0,\ldots,x_{k-1}}} \rangle = \langle F^3 \cup L^1 F^2 \rangle \cap \mathbb{B}[x_k, \ldots, x_n].$$

Proof. The output after eliminating $x_0, \ldots, x_{i-1}$ are the sets $F^3_{\overline{x_0,\ldots,x_{i-1}}}$ and $F^2_{\overline{x_0,\ldots,x_{i-1}}}$. These sets form the input for eliminating x_i, and applying Theorem 18 on these input sets gives

$$\langle F^3_{\overline{x_0,\ldots,x_i}} \cup L^1_{\overline{x_0,\ldots,x_i}} F^2_{\overline{x_0,\ldots,x_i}} \rangle = \langle F^3_{\overline{x_0,\ldots,x_{i-1}}} \cup L^{\leq 1}_{\overline{x_0,\ldots,x_{i-1}}} F^2_{\overline{x_0,\ldots,x_{i-1}}} \rangle \cap \mathbb{B}[x_{i+1}, \ldots, x_n],$$

for each i in $\{0, \ldots, k-1\}$. Substituting these equations into each other creates the stated relation between the original F^3, F^2 and $F^3_{\overline{x_0,\ldots,x_{k-1}}}, F^2_{\overline{x_0,\ldots,x_{k-1}}}$. ■

4.2. Degree bound $d \geq 4$

We explain here how the elimination procedure needs to be generalized when we increase the degree bound to some $d \geq 4$. The construction is essentially the same as in the previous section, but there will be more syzygies than Koszul and Boolean syzygies, as pointed out in Theorem 16.

Elimination procedure:

1. We split the set F^2 into subsets $F^{2,norm}_{x_0}$ containing x_0 and $F^2_{\overline{x_0}}$ not containing x_0 by using *SplitVariable*(F^2, x_0). These sets can be used to increase F^3, by adding $(x_0+1)F^{2,norm}_{x_0}$ and $x_0 F^2_{\overline{x_0}}$ to F^3.
2. Normalize F^3 with respect to $F^{2,norm}_{x_0}$, producing $F^{3,norm}_{x_0}$ and $F^3_{\overline{x_0}}$.
3. Compute the resultants and coefficient constraints from $F^{2,norm}_{x_0}$ and add to $F^3_{\overline{x_0}}$
4. Add the sets $(x_0+1)F^{3,norm}_{x_0}$ and $x_0 F^3_{\overline{x_0}}$ to F^4.
5. Normalize F^4 with respect to $F^{3,norm}_{x_0} \cup F^{2,norm}_{x_0}$, giving the sets $F^{4,norm}_{x_0}$ and $F^4_{\overline{x_0}}$.
6. Compute resultants and coefficient constraints of the sets $F^{3,norm}_{x_0}$ and $F^{2,norm}_{x_0}$ that respects the degree bound $d = 4$, and add these to $F^4_{\overline{x_0}}$ or $F^3_{\overline{x_0}}$, according to degree.
7. Compute the syzygies R^3 of degree 3 from $F^{4,norm}_{x_0} \cup F^{3,norm}_{x_0}$. These generate a set T^4 of polynomials of degree ≤ 4 in $\mathbb{B}[x_1, \ldots, x_n]$. Add T_4 to $F^4_{\overline{x_0}}$.

The outputs of the procedure are the sets $F^2_{\overline{x_0}}, F^3_{\overline{x_0}}$ and $F^4_{\overline{x_0}}$. Concerning Step 6 let $x_0 a_i^2 + b_i^3$ be the polynomials in $F^{3,norm}_{x_0}$ and $x_0 a_i^3 + b_i^4$ be those in $F^{4,norm}_{x_0}$. We need to compute the syzygies of degree 3

$$\sum_i r_i^1 \varepsilon_i^2 + \sum_i r_i^0 \varepsilon_i^3$$

(so the r_i^0 are constants) giving relations between the a_i^2 and the a_i^3. Then T^4 consists of the polynomials $\sum_i r_i^1 b^3 + \sum_i r_i^0 b_i^4$ which are of degree ≤ 4. Note that these are in $\mathbb{B}[x_1, \cdots, x_n]$.

Theorem 20. *Let F^2, F^3, F^4 be the input set of polynomials, and $F^2_{\overline{x_0}}, F^3_{\overline{x_0}}$ and $F^4_{\overline{x_0}}$ be the outputs of the elimination procedure.*

a. The linear span $\langle F^4 \cup L^1 F^3 \cup L^2 F^2 \rangle$ can be expressed as

$$\langle F^{4,norm}_{x_0} \cup L^1_{\overline{x_0}} F^{3,norm}_{x_0} \cup L^2_{\overline{x_0}} F^{2,norm}_{x_0} \rangle + \langle F^4_{\overline{x_0}} \cup L^1_{\overline{x_0}} F^3_{\overline{x_0}} \cup L^2_{x_0} F^2_{\overline{x_0}} \rangle$$

b. The elimination space

$$\langle F^4 \cup L^1 F^3 \cup L^2 F^2 \rangle \cap \mathbb{B}[x_1, \ldots, x_n] = \langle F^4_{\overline{x_0}} \cup L^1_{\overline{x_0}} F^3_{\overline{x_0}} \cup L^2_{x_0} F^2_{\overline{x_0}} \rangle$$

Proof. a. The inclusion $\supseteq$ is clear. We shall prove $\subseteq$. First write

$$\langle F^4 \cup L^1F^3 \cup L^2F^2\rangle = \langle F^4\rangle + \langle L^1\rangle \cdot \langle F^3 \cup L^1F^2\rangle. \tag{11}$$

By the previous Theorem 18a. we may express the right side above as:

$$\langle F^4\rangle + \langle L^1\rangle\langle F^{3,norm}_{x_0} \cup L^1_{\overline{x_0}}F^{2,norm}_{x_0} \cup F^3_{\overline{x_0}} \cup L^1_{\overline{x_0}}F^2_{\overline{x_0}}\rangle.$$

We can split $\langle L^1\rangle$ as

$$\langle x_0+1\rangle + \langle L^1_{\overline{x_0}}\rangle, \text{ or } \langle x_0\rangle + \langle L^1_{\overline{x_0}}\rangle.$$

Hence the above is :

$$\begin{aligned} & F^4 + (x_0+1)F^{3,norm}_{x_0} + (x_0+1)\langle L^1_{\overline{x_0}}F^{2,norm}_{x_0}\rangle + x_0\langle F^3_{\overline{x_0}}\rangle + x_0\langle L^1_{x_0}F^2_{\overline{x_0}}\rangle \\ + \quad & \langle L^1_{x_0}\rangle\langle F^{3,norm}_{\overline{x_0}} \cup F^3_{\overline{x_0}} \cup L^1_{\overline{x_0}}F^{2,norm}_{x_0} \cup L^1_{\overline{x_0}}F^2_{\overline{x_0}}\rangle. \end{aligned}$$

Let us look at the various parts of this expression. By the algorithm :

$$(x_0+1)\langle F^{2,norm}\rangle + x_0\langle F^2_{\overline{x_0}}\rangle \subseteq \langle F^{3,norm}_{x_0} \cup F^3_{\overline{x_0}} \cup L^1_{\overline{x_0}}F^{2,norm}_{x_0}\rangle.$$

Hence

$$\langle L^1_{x_0}\rangle \cdot ((x_0+1)\langle F^{2,norm}\rangle + x_0\langle F^2_{\overline{x_0}}\rangle) \subseteq \langle L^1_{\overline{x_0}}F^{3,norm}_{x_0} \cup L^1_{\overline{x_0}}F^3_{\overline{x_0}} \cup L^2_{\overline{x_0}}F^{2,norm}_{x_0}\rangle.$$

Also by the algorithm:

$$F^4 + (x_0+1)\langle F^{3,norm}_{x_0}\rangle + x_0\langle F^3_{\overline{x_0}}\rangle \subseteq \langle F^{4,norm} \cup F^4_{\overline{x_0}} \cup L^1_{\overline{x_0}}F^{3,norm}_{x_0} \cup L^2_{\overline{x_0}}F^{2,norm}_{x_0}\rangle.$$

The upshot of this is that the left side of (11) is contained in

$$\langle F^{4,norm}_{x_0} \cup F^4_{\overline{x_0}} \cup L^1_{\overline{x_0}}F^{3,norm}_{x_0} \cup L^1_{\overline{x_0}}F^3_{x_0} \cup L^2_{\overline{x_0}}F^{2,norm} \cup L^2_{\overline{x_0}}F^2_{\overline{x_0}}\rangle,$$

and this proves part a.

b. By part a. above this will follow if we can show that

$$\langle F^{4,norm}_{x_0} \cup L^1_{\overline{x_0}}F^{3,norm}_{x_0} \cup L^2_{\overline{x_0}}F^{2,norm}_{x_0}\rangle \cap \mathbb{B}[x_1,\ldots,x_n] \tag{12}$$

is contained in

$$\langle F^4_{\overline{x_0}} \cup L^1_{\overline{x_0}}F^3_{\overline{x_0}} \cup L^2_{x_0}F^2_{\overline{x_0}}\rangle.$$

If we write $x_0a_i^{d-1} + b_i^d$ for the polynomials in $F^{d,norm}_{x_0}$, we are looking at expressions

$$\sum_i q_i(x_0a_i^1 + b_i^2) + \sum_i l_i(x_0a_i^2 + b_i^3) + \sum_i c_i(x_0a_i^3 + b_i^4) \tag{13}$$

where the q, l and c are respectively quadratic, linear and constants in $\mathbb{B}[x_1,\ldots,x_n]$, and this expression must be in $\mathbb{B}[x_1,\ldots,x_n]$. So we must have a relation

$$\sum_i q_ia_i^1 + \sum_i l_ia_i^2 + \sum_i c_ia_i^3 = 0.$$

By Theorem 16, this gives a syzygy expressible in terms of Koszul and Boolean sygygies, and the extra syzygies in R^3. Hence the expression (13) can be expressed in terms of resultants, coefficient constraints and the extra elements T_4 which have already been added to $F^4_{\overline{x_0}}$. This proves part b. ∎

If we increase to $d = 5$ or higher, the elimination procedure follows the same lines as for $d = 3$ and $d = 4$. We normalize the $F^i_{x_0}$ using the already normalized sets of lower-degree polynomials as basis, compute resultants and coefficient constraints and all other syzygies needed for eliminating x_0. When d grows, the space for possible non-trivial syzygies increases. A topic for further research is to investigate how large fraction of all syzygies that are covered by Boolean and Koszul syzygies when d increases.

5. Conclusions

In this paper we have studied how to eliminate variables from Boolean equation systems when the degree of polynomials produced is upper bounded. The tools we use for elimination are the known techniques of *normalization, resultants*. In addition we introduced *coefficient constraints*, which only applies for characteristic 2. We have explained how to modify the tools to satisfy the upper bound on the degree.

The motivation for our study comes from solving non-linear Boolean equation systems. The most well-known algorithms to solve such systems are focused on Gröbner bases (F4/F5) or relinearization after multiplying all polynomials with a set of monomials (XL). We relate the work in this paper to these approaches. What we found is that our elimination tools may be explained in terms of Gröbner basis algorithms, but where we only reduce an S-polynomial modulo a sub-basis and stop the reduction as soon as the variable in question has been eliminated. With regard to XL methods, we found that our elimination procedure will compute the same output as multiplying all polynomials with all allowed monomials and doing Gaussian elimination to remove a variable from the system. The elimination procedure presented in this paper is a lot more efficient though, since we do not multiply with all monomials but only compute exactly what is needed to eliminate the desired variable.

Eliminating variables may be explained in terms of syzygies, where Koszul syzygies correspond to resultants and Boolean syzygies correspond to coefficient constraints. We found that when the polynomials the syzygies are computed for are all linear, all syzygies can be generated by the syzygies of the Koszul and Boolean types.

In order to limit the length of this paper, we have avoided the discussion of several closely related questions. One topic for further research is to develop algorithms for solving non-linear Boolean equation systems based on variable elimination. Another is to investigate how much of the complete syzygy space is generated by the trivial Boolean and Koszul syzygies, or, in other words, how rare nontrivial syzygies are. Yet another line of investigation should examine and quantify the loss of information associated with the limitation of the degree.

References

[1] M. Albrecht, C. Rechberger, T. Schneider, T. Tiessen, M. Zohner. *Ciphers for MPC and FHE,* Eurocrypt 2015, LNCS 9056, pp. 430 – 454, Springer, 2015.

[2] D.Cox, J.Little, D.O'Shea, *Ideals, varieties and algorithms*, Third edition, 2007 Springer Science and Business Media.

[3] D.Cox, J.Little, D.O'Shea *Using Algebraic Geometry* GTM 185, Springer Science and Business Media 2005.

[4] D Lazard. *Gaussian elimination and resolution of systems of algebraic equations.* In proc. EUROCAL 1983, vol 162 of Lect. Notes in Comp. Sci, pp. 146-157.

[5] Kipnis A., Shamir A. *Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization.* Advances in Cryptology — CRYPTO' 99. CRYPTO 1999. Lecture Notes in Computer Science, vol 1666, pp. 19 – 30. Springer, Berlin, Heidelberg 1999.

[6] A. Shamir, J. Patarin, N. Courtois, A. Klimov, *Efficient Algorithms for solving Overdefined Systems of Multivariate Polynomial Equations*, Eurocrypt'2000, LNCS 1807, pp. 392 — 407, Springer 2000.

[7] Courtois N.T., Pieprzyk J. *Cryptanalysis of Block Ciphers with Overdefined Systems of Equations*, Advances in Cryptology — ASIACRYPT 2002. ASIACRYPT 2002. Lecture Notes in Computer Science, vol 2501, pp. 267 – 287. Springer, Berlin, Heidelberg 2002

[8] Murphy S., Robshaw M.J. *Essential Algebraic Structure within the AES*. Advances in Cryptology — CRYPTO 2002. CRYPTO 2002. Lecture Notes in Computer Science, vol 2442, pp. 1 – 16. Springer, Berlin, Heidelberg 2002

[9] Biryukov A., De Cannière C. *Block Ciphers and Systems of Quadratic Equations*, Fast Software Encryption, FSE 2003. Lecture Notes in Computer Science, vol 2887, pp. 274 – 289. Springer, Berlin, Heidelberg 2003

[10] J-C. Faugere. *A new efficient algorithm for computing Gröbner bases (F4). Effective methods in algebraic geometry* (Saint-Malo, 1998), J. Pure Appl. Algebra 139 (1999)

[11] J-C. Faugere. *A new efficient algorithm for computing Gröbner bases without reduction to zero (F5)*, Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation, 75–83, ACM, New York, 2002.

[12] T. Stegers. *Faugere's F5 Algorithm Revisited*, Thesis For The Degree Of Diplom-Mathematiker, Department of Mathematics, Technische Universität Darmstadt, 2005. Available at http://sciencedocbox.com/Physics/68613748-Faugere-s-f5-algorithm-revisited.html

[13] M. Sala, T. Mora, L. Perret, S. Sakata, C. Traverso, *Gröbner Bases, Coding and Cryptography*, Springer, 2009