Re-linearization and elimination of variables in Boolean equation systems

Bjørn Møller Greve 1,2 Håvard Raddum 2 Øyvind Ytrehus 2

¹Norwegian Defence Research Establishment

²Simula@UiB

4 September, 2017





	Introduction previous work ●○○			Examples	simula@ui
--	-----------------------------------	--	--	----------	-----------

Elimination of variables from Boolean functions

- Consider the Boolean ring $B[1,n] = \mathbb{F}_2[x_1,\ldots,x_n]/(x_i^2+x_i|i=1,\ldots,n)$
- Eliminate x_1 s.th (a_1, \ldots, a_n) solution in left system $\implies (a_2, \ldots, a_n)$ is solution in right system.

- Describe cipher as quadratic Boolean equation system.
- Variables: Secret key K + auxiliary variables (To keep equations simple)
- Is it possible to eliminate auxiliary variables and find some equations in only key variables?

Elimination of variables from Boolean functions

- Consider the Boolean ring $B[1,n] = \mathbb{F}_2[x_1,\ldots,x_n]/(x_i^2+x_i|i=1,\ldots,n)$
- Eliminate x_1 s.th (a_1, \ldots, a_n) solution in left system $\implies (a_2, \ldots, a_n)$ is solution in right system.

- Describe cipher as quadratic Boolean equation system.
- Variables: Secret key K + auxiliary variables (To keep equations simple)
- Is it possible to eliminate auxiliary variables and find some equations in only key variables?

Elimination of variables from Boolean functions

• Consider the Boolean ring $B[1,n] = \mathbb{F}_2[x_1,\ldots,x_n]/(x_i^2+x_i|i=1,\ldots,n)$

• Eliminate x_1 s.th (a_1, \ldots, a_n) solution in left system $\implies (a_2, \ldots, a_n)$ is solution in right system.

- Describe cipher as quadratic Boolean equation system.
- Variables: Secret key K + auxiliary variables (To keep equations simple)
- Is it possible to eliminate auxiliary variables and find some equations in only key variables?

Elimination of variables from Boolean functions

- Consider the Boolean ring $B[1,n] = \mathbb{F}_2[x_1,\ldots,x_n]/(x_i^2+x_i|i=1,\ldots,n)$
- Eliminate x_1 s.th (a_1, \ldots, a_n) solution in left system $\implies (a_2, \ldots, a_n)$ is solution in right system.

- Describe cipher as quadratic Boolean equation system.
- Variables: Secret key K + auxiliary variables (To keep equations simple)
- Is it possible to eliminate auxiliary variables and find some equations in only key variables?

Elimination of variables from Boolean functions

- Consider the Boolean ring $B[1,n] = \mathbb{F}_2[x_1,\ldots,x_n]/(x_i^2+x_i|i=1,\ldots,n)$
- Eliminate x_1 s.th (a_1, \ldots, a_n) solution in left system $\implies (a_2, \ldots, a_n)$ is solution in right system.

- Describe cipher as quadratic Boolean equation system.
- Variables: Secret key K + auxiliary variables (To keep equations simple)
- Is it possible to eliminate auxiliary variables and find some equations in only key variables?

Elimination of variables from Boolean functions

- Consider the Boolean ring $B[1,n] = \mathbb{F}_2[x_1,\ldots,x_n]/(x_i^2+x_i|i=1,\ldots,n)$
- Eliminate x_1 s.th (a_1, \ldots, a_n) solution in left system $\implies (a_2, \ldots, a_n)$ is solution in right system.

Applications to ciphers

- Describe cipher as quadratic Boolean equation system.
- Variables: Secret key K + auxiliary variables (To keep equations simple)
- Is it possible to eliminate auxiliary variables and find some equations in only key variables?

Elimination of variables from Boolean functions

- Consider the Boolean ring $B[1,n] = \mathbb{F}_2[x_1,\ldots,x_n]/(x_i^2+x_i|i=1,\ldots,n)$
- Eliminate x_1 s.th (a_1, \ldots, a_n) solution in left system $\implies (a_2, \ldots, a_n)$ is solution in right system.

- Describe cipher as quadratic Boolean equation system.
- Variables: Secret key K + auxiliary variables (To keep equations simple)
- Is it possible to eliminate auxiliary variables and find some equations in only key variables?

The general method:

- Save intermediate systems after each elimination.
- Brute force possible solutions of final system, lift through intermediate systems to filter out false solutions.

The block cipher method:

Repeating the process of variable elimination using other known plaintext/ciphertext pairs and build up a low-degree system of equations in only user-selected key variables that has K as a unique solution.

Re-linearization

Solve by re-linearization if we can generate more linearly independent polynomials (in some acceptable degree) than there are monomials.

The general method:

- Save intermediate systems after each elimination.
- Brute force possible solutions of final system, lift through intermediate systems to filter out false solutions.

The block cipher method:

Repeating the process of variable elimination using other known plaintext/ciphertext pairs and build up a low-degree system of equations in only user-selected key variables that has K as a unique solution.

Re-linearization

Solve by re-linearization if we can generate more linearly independent polynomials (in some acceptable degree) than there are monomials.

The general method:

- Save intermediate systems after each elimination.
- Brute force possible solutions of final system, lift through intermediate systems to filter out false solutions.

The block cipher method:

Repeating the process of variable elimination using other known plaintext/ciphertext pairs and build up a low-degree system of equations in only user-selected key variables that has K as a unique solution.

Re-linearization

Solve by re-linearization if we can generate more linearly independent polynomials (in some acceptable degree) than there are monomials.

The general method:

- Save intermediate systems after each elimination.
- Brute force possible solutions of final system, lift through intermediate systems to filter out false solutions.

The block cipher method:

Repeating the process of variable elimination using other known plaintext/ciphertext pairs and build up a low-degree system of equations in only user-selected key variables that has K as a unique solution.

Re-linearization

Solve by re-linearization if we can generate more linearly independent polynomials (in some acceptable degree) than there are monomials.

The general method:

- Save intermediate systems after each elimination.
- Brute force possible solutions of final system, lift through intermediate systems to filter out false solutions.

The block cipher method:

Repeating the process of variable elimination using other known plaintext/ciphertext pairs and build up a low-degree system of equations in only user-selected key variables that has K as a unique solution.

Re-linearization

Solve by re-linearization if we can generate more linearly independent polynomials (in some acceptable degree) than there are monomials.

The general method:

- Save intermediate systems after each elimination.
- Brute force possible solutions of final system, lift through intermediate systems to filter out false solutions.

The block cipher method:

Repeating the process of variable elimination using other known plaintext/ciphertext pairs and build up a low-degree system of equations in only user-selected key variables that has K as a unique solution.

Re-linearization

Solve by re-linearization if we can generate more linearly independent polynomials (in some acceptable degree) than there are monomials.

The general method:

- Save intermediate systems after each elimination.
- Brute force possible solutions of final system, lift through intermediate systems to filter out false solutions.

The block cipher method:

Repeating the process of variable elimination using other known plaintext/ciphertext pairs and build up a low-degree system of equations in only user-selected key variables that has K as a unique solution.

Re-linearization

Solve by re-linearization if we can generate more linearly independent polynomials (in some acceptable degree) than there are monomials.

Introduction previous work ○○●		simula@uib

GRFY (BFA 2017)

Elimination algorithm with degree restriction $deg(f_i) \leq 3$.

"Naive" XL elimination

- Multiply each f_i with all monomials respecting degree restriction \Rightarrow New polynomial set F.
- Gaussian elimination on F eliminating all monomials containing x_1 .

Theorem

- GRFY elimination = XL elimination when restricting the degree to ≤ 3 .
- In general: Extended GRFY elimination \supset XL elimination when restricting the degree to ≤ 3 .
- In general extended GRFY elimination introduces less false solutions than the naive XL method when restricting the degree to ≤ 3 .

	Introduction previous work			Examples	simula@	uib
--	----------------------------	--	--	----------	---------	-----

GRFY (BFA 2017)

Elimination algorithm with degree restriction $deg(f_i) \leq 3$.

"Naive" XL elimination

- Multiply each f_i with all monomials respecting degree restriction \Rightarrow New polynomial set F.
- Gaussian elimination on F eliminating all monomials containing x_1 .

Theorem

- GRFY elimination = XL elimination when restricting the degree to ≤ 3 .
- In general: Extended GRFY elimination \supset XL elimination when restricting the degree to ≤ 3 .
- In general extended GRFY elimination introduces less false solutions than the naive XL method when restricting the degree to ≤ 3 .

Introduction previous work		simula@uib

GRFY (BFA 2017)

Elimination algorithm with degree restriction $deg(f_i) \leq 3$.

"Naive" XL elimination

- Multiply each f_i with all monomials respecting degree restriction \Rightarrow New polynomial set F.
- Gaussian elimination on F eliminating all monomials containing x_1 .

Theorem

- GRFY elimination = XL elimination when restricting the degree to ≤ 3 .
- In general: Extended GRFY elimination \supset XL elimination when restricting the degree to ≤ 3 .
- In general extended GRFY elimination introduces less false solutions than the naive XL method when restricting the degree to ≤ 3 .

Introduction previous work		simula@uib

GRFY (BFA 2017)

Elimination algorithm with degree restriction $deg(f_i) \leq 3$.

"Naive" XL elimination

- Multiply each f_i with all monomials respecting degree restriction \Rightarrow New polynomial set F.
- Gaussian elimination on F eliminating all monomials containing x_1 .

Theorem

- GRFY elimination = XL elimination when restricting the degree to ≤ 3 .
- In general: Extended GRFY elimination \supset XL elimination when restricting the degree to $\leq 3.$
- In general extended GRFY elimination introduces less false solutions than the naive XL method when restricting the degree to ≤ 3 .

Introduction previous work		simula@uib

GRFY (BFA 2017)

Elimination algorithm with degree restriction $deg(f_i) \leq 3$.

"Naive" XL elimination

- Multiply each f_i with all monomials respecting degree restriction \Rightarrow New polynomial set F.
- Gaussian elimination on F eliminating all monomials containing x₁.

Theorem

- GRFY elimination = XL elimination when restricting the degree to ≤ 3 .
- In general: Extended GRFY elimination \supset XL elimination when restricting the degree to ≤ 3 .
- In general extended GRFY elimination introduces less false solutions than the naive XL method when restricting the degree to ≤ 3 .

Introduction previous work		simula@uib

GRFY (BFA 2017)

Elimination algorithm with degree restriction $deg(f_i) \leq 3$.

"Naive" XL elimination

- Multiply each f_i with all monomials respecting degree restriction \Rightarrow New polynomial set F.
- Gaussian elimination on F eliminating all monomials containing x_1 .

Theorem

- GRFY elimination = XL elimination when restricting the degree to ≤ 3 .
- In general: Extended GRFY elimination \supset XL elimination when restricting the degree to $\leq 3.$
- In general extended GRFY elimination introduces less false solutions than the naive XL method when restricting the degree to ≤ 3 .

Introduction previous work		simula@uib

GRFY (BFA 2017)

Elimination algorithm with degree restriction $deg(f_i) \leq 3$.

"Naive" XL elimination

- Multiply each f_i with all monomials respecting degree restriction \Rightarrow New polynomial set F.
- Gaussian elimination on F eliminating all monomials containing x_1 .

Theorem

- GRFY elimination = XL elimination when restricting the degree to ≤ 3 .
- In general: Extended GRFY elimination \supset XL elimination when restricting the degree to $\leq 3.$
- In general extended GRFY elimination introduces less false solutions than the naive XL method when restricting the degree to ≤ 3 .

Generalization of Previous work		simula@uib

Main idea

- Allow more computational complexity when eliminating variables \rightarrow fixing the degree at a chosen parameter $d \ge 3$.
- $F^d = \{ \text{polynomials of deg } d \}, F^{d-1} = \{ \text{pols. of deg } d-1 \}, \dots, F^1 = \{ \text{linear polynomials} \}.$

- Eliminate x_1, \ldots , only computing with polynomials of degree d or less.
- $L^0 = \{1\}, L^1 = \{x_1, \dots, x_n\}, \dots, L^i = \{\text{monomials of degree } i\}.$
- Bounding degree $d \to$ form any product of the form $L^i F^j = \{lf, l \in L^i, f \in F^j\}$ as long as $i + j \leq d$.
- Eliminate variables from the vectorspace $\langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$

Generalization of Previous work		simula@uib
C	1	

Main idea

- Allow more computational complexity when eliminating variables \rightarrow fixing the degree at a chosen parameter $d \ge 3$.
- $F^d = \{ \text{polynomials of deg } d \}, F^{d-1} = \{ \text{pols. of deg } d-1 \}, \dots, F^1 = \{ \text{linear polynomials} \}.$

- Eliminate x_1, \ldots , only computing with polynomials of degree d or less.
- $L^0 = \{1\}, L^1 = \{x_1, \dots, x_n\}, \dots, L^i = \{\text{monomials of degree } i\}.$
- Bounding degree $d \to$ form any product of the form $L^i F^j = \{lf, l \in L^i, f \in F^j\}$ as long as $i + j \leq d$.
- Eliminate variables from the vectorspace $\langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \dots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$

Generalization of Previous work		simula@uib
0		

Main idea

- Allow more computational complexity when eliminating variables \rightarrow fixing the degree at a chosen parameter $d \ge 3$.
- $F^d = \{ \text{polynomials of deg } d \}, F^{d-1} = \{ \text{pols. of deg } d-1 \}, \dots, F^1 = \{ \text{linear polynomials} \}.$

- Eliminate x_1, \ldots , only computing with polynomials of degree d or less.
- $L^0 = \{1\}, L^1 = \{x_1, \dots, x_n\}, \dots, L^i = \{\text{monomials of degree } i\}.$
- Bounding degree $d \to$ form any product of the form $L^i F^j = \{lf, l \in L^i, f \in F^j\}$ as long as $i + j \leq d$.
- Eliminate variables from the vectorspace $\langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \dots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$

Generalization of Previous work		simula@uib
C	I	

Main idea

- Allow more computational complexity when eliminating variables \rightarrow fixing the degree at a chosen parameter $d \ge 3$.
- $F^d = \{ \text{polynomials of deg } d \}, F^{d-1} = \{ \text{pols. of deg } d-1 \}, \dots, F^1 = \{ \text{linear polynomials} \}.$

- Eliminate x_1, \ldots , only computing with polynomials of degree d or less.
- $L^0 = \{1\}$, $L^1 = \{x_1, \dots, x_n\}, \dots, L^i = \{\text{monomials of degree } i\}$.
- Bounding degree $d \to form$ any product of the form $L^i F^j = \{lf, l \in L^i, f \in F^j\}$ as long as $i + j \leq d$.
- Eliminate variables from the vectorspace $\langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \dots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$

Generalization of Previous work		simula@uib
C	1	

Main idea

- Allow more computational complexity when eliminating variables \rightarrow fixing the degree at a chosen parameter $d \ge 3$.
- $F^d = \{ \text{polynomials of deg } d \}, F^{d-1} = \{ \text{pols. of deg } d 1 \}, \dots, F^1 = \{ \text{linear polynomials} \}.$

- Eliminate x_1, \ldots , only computing with polynomials of degree d or less.
- $L^0 = \{1\}$, $L^1 = \{x_1, \dots, x_n\}, \dots, L^i = \{\text{monomials of degree } i\}$.
- Bounding degree $d \to \text{form}$ any product of the form $L^i F^j = \{lf, l \in L^i, f \in F^j\}$ as long as $i + j \leq d$.
- Eliminate variables from the vectorspace $\langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \dots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$

Generalization of Previous work		simula@uib
C	1	

Main idea

- Allow more computational complexity when eliminating variables \rightarrow fixing the degree at a chosen parameter $d \ge 3$.
- $F^d = \{ \text{polynomials of deg } d \}, F^{d-1} = \{ \text{pols. of deg } d 1 \}, \dots, F^1 = \{ \text{linear polynomials} \}.$

Objective

- Eliminate x_1, \ldots , only computing with polynomials of degree d or less.
- $L^0 = \{1\}, L^1 = \{x_1, ..., x_n\}, ..., L^i = \{\text{monomials of degree } i\}.$
- Bounding degree $d \to$ form any product of the form $L^i F^j = \{lf, l \in L^i, f \in F^j\}$ as long as $i + j \leq d$.

• Eliminate variables from the vector space $\langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$

Generalization of Previous work		simula@uib
C I		

Main idea

- Allow more computational complexity when eliminating variables \rightarrow fixing the degree at a chosen parameter $d \ge 3$.
- $F^d = \{ \text{polynomials of deg } d \}, F^{d-1} = \{ \text{pols. of deg } d-1 \}, \dots, F^1 = \{ \text{linear polynomials} \}.$

- Eliminate x_1, \ldots , only computing with polynomials of degree d or less.
- $L^0 = \{1\}, L^1 = \{x_1, ..., x_n\}, ..., L^i = \{\text{monomials of degree } i\}.$
- Bounding degree $d \to$ form any product of the form $L^i F^j = \{lf, l \in L^i, f \in F^j\}$ as long as $i + j \leq d$.
- Eliminate variables from the vectorspace $\langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \dots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$.

	Elimination techniques ●○○	simula@uib

A. "Naive" XL elimination Monomials containing x_1 are largest For each $i = \{1, ..., d\}$, Gaussian elimination on $F^i \cup L^1 F^{i-1} \cup ... \cup L^{i-2} F^2 \cup L^{i-1} F^1$ to eliminate all monomials containing x_1 .

B. Ordering the monomials with respect to degree

- For each $i = \{1 \dots, d\}$, $\langle F^i \cup L^1 F^{i-1} \cup \dots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$ may contain more polynomials of degree < i.
- We can try to produce a larger set of polynomials $F^{i-1,(2)}, \ldots, F^{1,(2)}$ by Gaussian elimination with respect to degree. I.e $F^{i-1} \subseteq F^{i-1,(2)}, \ldots, F^1 \subseteq F^{1,(2)}.$

- Enable us to eliminate particular monomials containing x_1 from each F^i using the lower degree sets $F^{i-1}, \ldots, F^2, F^1$ as basis.
- The effect of normalization is that there is a rather large set of monomials containing x_1 that can not appear in each set F^i at the end.

A. "Naive" XL elimination Monomials containing x_1 are largest For each $i = \{1..., d\}$, Gaussian elimination on $F^i \cup L^1 F^{i-1} \cup \ldots \cup L^{i-2} F^2 \cup L^{i-1} F^1$ to eliminate all monomials containing x_1 .

B. Ordering the monomials with respect to degree

- For each $i = \{1..., d\}$, $\langle F^i \cup L^1 F^{i-1} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$ may contain more polynomials of degree < i.
- We can try to produce a larger set of polynomials $F^{i-1,(2)}, \ldots, F^{1,(2)}$ by Gaussian elimination with respect to degree. I.e $F^{i-1} \subseteq F^{i-1,(2)}, \ldots, F^1 \subseteq F^{1,(2)}.$

- Enable us to eliminate particular monomials containing x_1 from each F^i using the lower degree sets $F^{i-1}, \ldots, F^2, F^1$ as basis.
- The effect of normalization is that there is a rather large set of monomials containing x_1 that can not appear in each set F^i at the end.

A. "Naive" XL elimination Monomials containing x_1 are largest For each $i = \{1..., d\}$, Gaussian elimination on $F^i \cup L^1 F^{i-1} \cup \ldots \cup L^{i-2} F^2 \cup L^{i-1} F^1$ to eliminate all monomials containing x_1 .

B. Ordering the monomials with respect to degree

- For each $i = \{1..., d\}$, $\langle F^i \cup L^1 F^{i-1} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$ may contain more polynomials of degree $\langle i$.
- We can try to produce a larger set of polynomials $F^{i-1,(2)}, \ldots, F^{1,(2)}$ by Gaussian elimination with respect to degree. I.e $F^{i-1} \subseteq F^{i-1,(2)}, \ldots, F^1 \subseteq F^{1,(2)}.$

- Enable us to eliminate particular monomials containing x_1 from each F^i using the lower degree sets $F^{i-1}, \ldots, F^2, F^1$ as basis.
- The effect of normalization is that there is a rather large set of monomials containing x_1 that can not appear in each set F^i at the end.

	Elimination techniques ●○○	simula@uib

A. "Naive" XL elimination Monomials containing x_1 are largest For each $i = \{1..., d\}$, Gaussian elimination on $F^i \cup L^1 F^{i-1} \cup \ldots \cup L^{i-2} F^2 \cup L^{i-1} F^1$ to eliminate all monomials containing x_1 .

- B. Ordering the monomials with respect to degree
 - For each $i = \{1..., d\}$, $\langle F^i \cup L^1 F^{i-1} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$ may contain more polynomials of degree < i.

• We can try to produce a larger set of polynomials $F^{i-1,(2)}, \ldots, F^{1,(2)}$ by Gaussian elimination with respect to degree. I.e $F^{i-1} \subseteq F^{i-1,(2)}, \ldots, F^1 \subseteq F^{1,(2)}$.

- Enable us to eliminate particular monomials containing x_1 from each F^i using the lower degree sets $F^{i-1}, \ldots, F^2, F^1$ as basis.
- The effect of normalization is that there is a rather large set of monomials containing x_1 that can not appear in each set F^i at the end.

	Elimination techniques ●○○	simula@uib

A. "Naive" XL elimination Monomials containing x_1 are largest For each $i = \{1..., d\}$, Gaussian elimination on $F^i \cup L^1 F^{i-1} \cup \ldots \cup L^{i-2} F^2 \cup L^{i-1} F^1$ to eliminate all monomials containing x_1 .

- B. Ordering the monomials with respect to degree
 - For each $i = \{1..., d\}$, $\langle F^i \cup L^1 F^{i-1} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$ may contain more polynomials of degree < i.
 - We can try to produce a larger set of polynomials $F^{i-1,(2)}, \ldots, F^{1,(2)}$ by Gaussian elimination with respect to degree. I.e $F^{i-1} \subseteq F^{i-1,(2)}, \ldots, F^1 \subseteq F^{1,(2)}$.

- Enable us to eliminate particular monomials containing x_1 from each F^i using the lower degree sets $F^{i-1}, \ldots, F^2, F^1$ as basis.
- The effect of normalization is that there is a rather large set of monomials containing x_1 that can not appear in each set F^i at the end.

	Elimination techniques ●○○	simula@uib

A. "Naive" XL elimination Monomials containing x_1 are largest For each $i = \{1..., d\}$, Gaussian elimination on $F^i \cup L^1 F^{i-1} \cup \ldots \cup L^{i-2} F^2 \cup L^{i-1} F^1$ to eliminate all monomials containing x_1 .

- B. Ordering the monomials with respect to degree
 - For each $i = \{1..., d\}$, $\langle F^i \cup L^1 F^{i-1} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$ may contain more polynomials of degree < i.
 - We can try to produce a larger set of polynomials $F^{i-1,(2)}, \ldots, F^{1,(2)}$ by Gaussian elimination with respect to degree. I.e $F^{i-1} \subseteq F^{i-1,(2)}, \ldots, F^1 \subseteq F^{1,(2)}$.

- Enable us to eliminate particular monomials containing x_1 from each F^i using the lower degree sets $F^{i-1}, \ldots, F^2, F^1$ as basis.
- The effect of normalization is that there is a rather large set of monomials containing x_1 that can not appear in each set F^i at the end.

	Elimination techniques ●○○	simula@uib

A. "Naive" XL elimination Monomials containing x_1 are largest For each $i = \{1..., d\}$, Gaussian elimination on $F^i \cup L^1 F^{i-1} \cup \ldots \cup L^{i-2} F^2 \cup L^{i-1} F^1$ to eliminate all monomials containing x_1 .

- B. Ordering the monomials with respect to degree
 - For each $i = \{1..., d\}$, $\langle F^i \cup L^1 F^{i-1} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$ may contain more polynomials of degree < i.
 - We can try to produce a larger set of polynomials $F^{i-1,(2)}, \ldots, F^{1,(2)}$ by Gaussian elimination with respect to degree. I.e $F^{i-1} \subseteq F^{i-1,(2)}, \ldots, F^1 \subseteq F^{1,(2)}$.

- Enable us to eliminate particular monomials containing x_1 from each F^i using the lower degree sets $F^{i-1}, \ldots, F^2, F^1$ as basis.
- The effect of normalization is that there is a rather large set of monomials containing x_1 that can not appear in each set F^i at the end.

	Elimination techniques ●○○	simula@uib

The monomial orders

A. "Naive" XL elimination Monomials containing x_1 are largest For each $i = \{1..., d\}$, Gaussian elimination on $F^i \cup L^1 F^{i-1} \cup \ldots \cup L^{i-2} F^2 \cup L^{i-1} F^1$ to eliminate all monomials containing x_1 .

- B. Ordering the monomials with respect to degree
 - For each $i = \{1..., d\}$, $\langle F^i \cup L^1 F^{i-1} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$ may contain more polynomials of degree < i.
 - We can try to produce a larger set of polynomials $F^{i-1,(2)}, \ldots, F^{1,(2)}$ by Gaussian elimination with respect to degree. I.e $F^{i-1} \subseteq F^{i-1,(2)}, \ldots, F^1 \subseteq F^{1,(2)}$.

Normal forms

- Enable us to eliminate particular monomials containing x_1 from each F^i using the lower degree sets $F^{i-1}, \ldots, F^2, F^1$ as basis.
- The effect of normalization is that there is a rather large set of monomials containing x_1 that can not appear in each set F^i at the end.

	Elimination techniques ○●○	simula@uib

Resultants

- Given $f_i = a_i x_1 + b_i \in F^z$ and $f_j = a_j x_1 + b_j \in F^y$ satisfying $z + y \le d + 1$, where $\deg a_i \le z - 1$ and $\deg b_i \le z$ (resp j)
- We can form the resultant with respect to x_1

$$\operatorname{Res}(f_i, f_j) = a_i b_j + a_j b_i = a_i f_j + a_j f_i \in B[2, n]$$

• The set of all resultants: $\operatorname{Res}_2^{y+z} = \{\operatorname{Res}(f_i, f_j)\}.$

Coefficient constraints (GRFY 2017)

- Given $f = x_1 a + b \in F^i$ satisfying $2i \le d + 1$, where $\deg a \le i 1$ and $\deg b \le i$.
- We can form the coefficient constraint with respect to x_1

$$(a+1)f = x_1a(a+1) + b(a+1) = b(a+1) \in B[2,n].$$

	Elimination techniques ○●○	simula@uib

Resultants

- Given $f_i = a_i x_1 + b_i \in F^z$ and $f_j = a_j x_1 + b_j \in F^y$ satisfying $z + y \le d + 1$, where deg $a_i \le z - 1$ and deg $b_i \le z$ (resp j)
- We can form the resultant with respect to x_1

$$\operatorname{Res}(f_i, f_j) = a_i b_j + a_j b_i = a_i f_j + a_j f_i \in B[2, n]$$

• The set of all resultants: $\operatorname{Res}_2^{y+z} = \{\operatorname{Res}(f_i, f_j)\}.$

Coefficient constraints (GRFY 2017)

- Given $f = x_1 a + b \in F^i$ satisfying $2i \le d + 1$, where $\deg a \le i 1$ and $\deg b \le i$.
- We can form the coefficient constraint with respect to x_1

$$(a+1)f = x_1a(a+1) + b(a+1) = b(a+1) \in B[2,n].$$

	Elimination techniques ○●○	simula@uib

Resultants

- Given $f_i = a_i x_1 + b_i \in F^z$ and $f_j = a_j x_1 + b_j \in F^y$ satisfying $z + y \le d + 1$, where deg $a_i \le z - 1$ and deg $b_i \le z$ (resp j)
- We can form the resultant with respect to x_1

$$\operatorname{Res}(f_i, f_j) = a_i b_j + a_j b_i = a_i f_j + a_j f_i \in B[2, n]$$

• The set of all resultants: $\operatorname{Res}_2^{y+z} = \{\operatorname{Res}(f_i, f_j)\}.$

Coefficient constraints (GRFY 2017)

- Given $f = x_1a + b \in F^i$ satisfying $2i \le d + 1$, where $\deg a \le i 1$ and $\deg b \le i$.
- We can form the coefficient constraint with respect to x_1

$$(a+1)f = x_1a(a+1) + b(a+1) = b(a+1) \in B[2,n].$$

	Elimination techniques ○●○	simula@uib

Resultants

- Given $f_i = a_i x_1 + b_i \in F^z$ and $f_j = a_j x_1 + b_j \in F^y$ satisfying $z + y \le d + 1$, where deg $a_i \le z - 1$ and deg $b_i \le z$ (resp j)
- We can form the resultant with respect to x_1

$$\operatorname{Res}(f_i, f_j) = a_i b_j + a_j b_i = a_i f_j + a_j f_i \in B[2, n]$$

• The set of all resultants: $\operatorname{Res}_2^{y+z} = \{\operatorname{Res}(f_i, f_j)\}.$

Coefficient constraints (GRFY 2017)

- Given $f = x_1 a + b \in F^i$ satisfying $2i \le d + 1$, where $\deg a \le i 1$ and $\deg b \le i$.
- We can form the coefficient constraint with respect to x_1

$$(a+1)f = x_1a(a+1) + b(a+1) = b(a+1) \in B[2,n].$$

	Elimination techniques ○●○	simula@uib

Resultants

- Given $f_i = a_i x_1 + b_i \in F^z$ and $f_j = a_j x_1 + b_j \in F^y$ satisfying $z + y \le d + 1$, where deg $a_i \le z - 1$ and deg $b_i \le z$ (resp j)
- We can form the resultant with respect to x_1

$$\operatorname{Res}(f_i, f_j) = a_i b_j + a_j b_i = a_i f_j + a_j f_i \in B[2, n]$$

• The set of all resultants: $\operatorname{Res}_2^{y+z} = \{\operatorname{Res}(f_i, f_j)\}.$

Coefficient constraints (GRFY 2017)

- Given $f = x_1 a + b \in F^i$ satisfying $2i \le d + 1$, where $\deg a \le i 1$ and $\deg b \le i$.
- We can form the coefficient constraint with respect to x_1

$$(a+1)f = x_1a(a+1) + b(a+1) = b(a+1) \in B[2,n].$$

	Elimination techniques ○●○	simula@uib

Resultants

- Given $f_i = a_i x_1 + b_i \in F^z$ and $f_j = a_j x_1 + b_j \in F^y$ satisfying $z + y \le d + 1$, where deg $a_i \le z - 1$ and deg $b_i \le z$ (resp j)
- We can form the resultant with respect to x_1

$$\operatorname{Res}(f_i, f_j) = a_i b_j + a_j b_i = a_i f_j + a_j f_i \in B[2, n]$$

• The set of all resultants: $\operatorname{Res}_2^{y+z} = \{\operatorname{Res}(f_i, f_j)\}.$

Coefficient constraints (GRFY 2017)

- Given $f = x_1 a + b \in F^i$ satisfying $2i \le d + 1$, where $\deg a \le i 1$ and $\deg b \le i$.
- We can form the coefficient constraint with respect to x_1

$$(a+1)f = x_1a(a+1) + b(a+1) = b(a+1) \in B[2,n].$$

	Elimination techniques ○●○	simula@uib

Resultants

- Given $f_i = a_i x_1 + b_i \in F^z$ and $f_j = a_j x_1 + b_j \in F^y$ satisfying $z + y \le d + 1$, where deg $a_i \le z - 1$ and deg $b_i \le z$ (resp j)
- We can form the resultant with respect to x_1

$$\operatorname{Res}(f_i, f_j) = a_i b_j + a_j b_i = a_i f_j + a_j f_i \in B[2, n]$$

• The set of all resultants: $\operatorname{Res}_2^{y+z} = \{\operatorname{Res}(f_i, f_j)\}.$

Coefficient constraints (GRFY 2017)

- Given $f = x_1 a + b \in F^i$ satisfying $2i \le d + 1$, where $\deg a \le i 1$ and $\deg b \le i$.
- We can form the coefficient constraint with respect to x_1

$$(a+1)f = x_1a(a+1) + b(a+1) = b(a+1) \in B[2,n].$$

Theorem 1

- 1. {Resultants + coefficient constraints + Normalization ++} = $\langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle \cap B[2, n].$
- 2. If we extend the above construction to include **B**., we in general have {Resultants + coefficient constraints + Normalization ++} $\supset \langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle \cap B[2, n].$

In general we expect that we can eliminate variables with lower (monomial) complexity with generalized GRFY framework \rightarrow avoids multiplying with *all* variables.

Theorem 1

1. {Resultants + coefficient constraints + Normalization ++} = $\langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle \cap B[2,n].$

2. If we extend the above construction to include **B**., we in general have {Resultants + coefficient constraints + Normalization ++} $\supset \langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle \cap B[2, n].$

In general we expect that we can eliminate variables with lower (monomial) complexity with generalized GRFY framework \rightarrow avoids multiplying with *all* variables.

Theorem 1

- 1. {Resultants + coefficient constraints + Normalization ++} = $\langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle \cap B[2, n].$
- 2. If we extend the above construction to include **B**., we in general have {Resultants + coefficient constraints + Normalization ++} $\supset \langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle \cap B[2, n].$

In general we expect that we can eliminate variables with lower (monomial) complexity with generalized GRFY framework \rightarrow avoids multiplying with *all* variables.

Theorem 1

- 1. {Resultants + coefficient constraints + Normalization ++} = $\langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle \cap B[2, n].$
- 2. If we extend the above construction to include **B**., we in general have {Resultants + coefficient constraints + Normalization ++} $\supset \langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle \cap B[2, n].$

In general we expect that we can eliminate variables with lower (monomial) complexity with generalized GRFY framework \rightarrow avoids multiplying with *all* variables.

Theorem 1

- 1. {Resultants + coefficient constraints + Normalization ++} = $\langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle \cap B[2, n].$
- 2. If we extend the above construction to include **B**., we in general have {Resultants + coefficient constraints + Normalization ++} $\supset \langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle \cap B[2, n].$

In general we expect that we can eliminate variables with lower (monomial) complexity with generalized GRFY framework \rightarrow avoids multiplying with *all* variables.

	Examples	simula@uib

Random system, 8 equations in variables x_0, \ldots, x_7 , 1 unique solution

 $1 + x1^*x0 + x1 + x2 + x3^*x0 + x3^*x1 + x3^*x2 + x3 + x4^*x0 + x4^*x2 + x4 + x5^*x1 + x5^*x4 + x7^*x0 + x7^*x2 + x7^*x3 + x7^*x5 + x7^*$

x0 + x2 * x1 + x2 + x3 * x0 + x3 * x1 + x3 * x2 + x3 + x4 * x0 + x4 * x2 + x4 * x3 + x4 + x6 * x0 + x6 * x3 + x6 * x4 + x7 * x0 + x7 * x3 + x7 + x7 * x0 + x7 * x1 + x7 * x0 + x7 * x3 + x7 + x7 * x0 + x7 * x1 + x7 * x0 + x7 * x3 + x7 + x7 * x1 +

 $1 + x1 + x2^{*}x1 + x2 + x3^{*}x1 + x4^{*}x2 + x4 + x5^{*}x0 + x5^{*}x1 + x5^{*}x2 + x5 + x6^{*}x1 + x6^{*}x2 + x6^{*}x3 + x6^{*}x5 + x6 + x7^{*}x3 + x7^{*}x6 + x7^{*}x7 + x7^{*}x7 + x7^{*}x7 + x7^{*}x6 + x7^{*}x7 + x7$

x2+x3*x2+x4*x1+x4*x3+x4+x5*x2+x5*x3+x6*x1+x6*x2+x6*x3+x6*x4+x6*x5+x6+x7*x0+x7*x1+x7

1+x2*x1+x2+x3*x2+x4*x1+x4*x2+x4*x3+x5*x4+x5+x6*x2+x6*x3+x7*x5+x7

1+x0+x3*x2+x3+x4*x2+x5*x1+x5*x4+x5+x6*x2+x6*x3+x6*x5+x7*x4+x7*x6+x7

	Examples	simula@uib

** Restricting degree to max. 3 ** After elimination of M, got 7 polymonials: 15/56/7 * x26x1 * x26x6 * x46x5 * x26x6 * x26x6 * x16x6 * x16x6 * x26x6 * x16x6 * x16x5 * x16x5 * x16x7 * x5x7 * x5x7 * x5x7 * x5x6 * x166 * x266 * x26 * x16x6 * x16x * x16x

** Restricting degree to max. 4 **

After elimination of x0, got 14 polynomials:

x4x5x6x7 + x1x5x6x7 + x2x4x6x7 + x2x3x6x7 + x1x3x6x7 + x1x3x6x7 + x1x4x5x7 + x1x4x5x7 + x1x3x5x7 + x1x2x5x7 + x1x3x4x7 + x3x4x5x6 + x1x2x5x6 + x1x2x5x6 + x1x2x3x6 + x1x2x6 + x1x2x3x6 + x1x2x7 + x1x2x6 + x1x2x7 x3x4x5 + x1x2x3x4 + x5x6x7 + x3x6x7 + x4x5x7 + x2x5x7 + x1x5x7 + x3x4x7 + x2x3x7 + x1x3x7 + x4x5x6 + x2x5x6 + x1x5x6 + x2x4x6 + x2x3x6 + x1x3x6 + x3x4x5 + x1x3x5 + x1x3x5 + x1x3x5 + x1x2x 5 + x2x3x4 + x1x2x3 + x2x7 + x5x6 + x3x6 + x2x5 + x7 + x5 + x3 + x2 + 1 x3x5x6x7 + x1x5x6x7 + x2x4x6x7 + x2x3x6x7 + x1x2x6x7 + x3x4x5x7 + x1x2x4x7 + x1x2x4x7 + x1x2x3x7 + x2x4x5x6 + x1x3x4x6 + x1x3x4x6 + x1x3x4x5 + x1x2x4x5 + x1x2x3x5 + x3x5x7 + x3x4x5 + x1x2x4x7 + x1x2x7 + x1x7 + x1x2x7 + x1x7 + x1x x7 + x2x4x7 + x2x3x7 + x2x4x5 + x1x4x5 + x1x3x5 + x2x3x4 + x1x3x4 + x6x7 + x1x7 + x5x6 + x4x6 + x2x6 + x1x6 + x3x5 + x1x5 + x2x4 + x1x4 + x1x3 + x7 + x5 + x3 + x1 + 1 x2x5x6x7 + x1x5x6x7 + x1x4x6x7 + x2x4x5x7 + x1x4x5x7 + x1x2x5x7 + x2x3x4x7 + x2x3x5x6 + x1x3x5x6 + x1x3x4x6 + x1x2x4x6 + x2x3x4x5 + x1x2x3x4 + x5x6x7 + x2x6x7 + x2x6x7 + x2x5x7 + x2x5 x1x5x7 + x2x4x7 + x2x3x7 + x3x5x6 + x2x4x6 + x2x3x6 + x1x3x5 + x1x2x5 + x2x3x4 + x1x3x4 + x1x2x4 + x6x7 + x5x7 + x4x7 + x2x7 + x4x6 + x3x6 + x2x6 + x3x5 + x2x5 + x1x5 + x1x4 + x 7 + x6 + x5 + x3 + x2 + 1x1x5x6x7 + x2x3x6x7 + x1x3x5x7 + x1x3x5x7 + x1x2x3x7 + x2x3x5x6 + x1x2x5x6 + x1x3x4x6 + x1x3x4x5 + x1x2x3x4 + x2x6x7 + x2x3x7 + x1x3x7 + x1x2x7 + x2x5x6 + x1x5x6 + x1x3x4x6 + x 6 + x1x2x6 + x1x4x5 + x2x3x5 + x1x3x5 + x1x2x5 + x1x2x4 + x1x2x4 + x1x2x3 + x4x7 + x3x7 + x1x7 + x1x6 + x4x5 + x3x5 + x2x5 + x2x4 + x1x4 + x1x3 + x1x2 + x6 + x2 + x1 x3x4x6x7 + x1x4x6x7 + x1x2x6x7 + x3x4x5x7 + x1x4x5x7 + x2x3x5x7 + x1x2x5x7 + x1x2x4x7 + x1x2x4x7 + x1x2x3x7 + x3x4x5x6 + x1x2x4x6 + x1x2x4x6 + x1x2x4x6 + x1x3x4x5 + x1x2x4x6 + x1x2x6x6 + x1x2x6 + x1x2x6 + x1x2x6x6 + x1x2x6 + x1x2x6 + x1x2x6x6 + x1x2x6 + x2x3x4 + x4x6x7 + x1x6x7 + x2x5x7 + x3x4x7 + x2x4x7 + x1x4x7 + x2x3x7 + x4x5x6 + x2x4x6 + x2x3x6 + x1x3x6 + x1x2x6 + x2x4x5 + x2x3x4 + x1x2x3 + x6x7 + x5x7 + x5x6 + x3x6 + x2x6 x1x6 + x3x5 + x2x4 + x7 + x5 + x3 + 1x2x4x6x7 + x1x3x6x7 + x1x2x6x7 + x1x3x4x7 + x1x2x4x7 + x3x4x5x6 + x2x4x5x6 + x1x4x5x6 + x2x3x5x6 + x1x2x5x6 + x1x2x5x6 + x1x2x3x6 + x2x3x4x5 + x5x6x7 + x4x6x7 + x3x6x7 + x2x6x7 + x2x6x7 + x2x6x7 + x2x6x7 + x3x6x7 + x2x6x7 + x3x6x7 + x3x7x7 + x3x7x7 + x3x7x7 + x3x7x7 + x3x7 x1x6x7 + x4x5x7 + x3x4x7 + x1x4x7 + x1x3x7 + x1x2x7 + x4x5x6 + x2x5x6 + x1x5x6 + x3x4x6 + x2x4x6 + x1x2x6 + x2x4x5 + x1x4x5 + x2x3x5 + x1x3x5 + x1x2x5 + x2x3x4 + x1x3x7 + x1x2x4 + x6x7 + x5x7 + x4x7 + x3x7 + x2x7 + x1x7 + x5x6 + x4x6 + x3x6 + x1x6 + x4x5 + x2x5 + x1x4 + x1x3 + x1x2 + x7 + x5 + x3 + x2 + x1 + 1x1x4x6x7 + x2x3x6x7 + x1x3x6x7 + x1x2x6x7 + x3x4x5x7 + x1x4x5x7 + x2x3x5x7 + x1x3x5x7 + x1x3x4x7 + x1x2x4x7 + x1x4x5x6 + x2x3x5x6 + x1x2x5x6 + x1x2x4x6 + x1x2x3x6 + x2x3x4x5 + x1 x3x4x5 + x1x2x3x4 + x5x6x7 + x2x6x7 + x4x5x7 + x2x5x7 + x3x4x7 + x1x3x7 + x1x2x7 + x3x5x6 + x1x5x6 + x3x4x6 + x2x3x6 + x1x3x6 + x2x4x5 + x1x3x5 + x1x2x5 + x1x2x5 + x1x2x4 + x1x2x3 + x6x7 + x3x7 + x1x6 + x4x5 + x3x5 + x3x4 + x2x4 + x2x3 + x1x2 + x4x5x6x7 + x4x6x7 + x2x6x7 + x1x6x7 + x1x5x7 + x1x5x7 + x3x4x7 + x2x4x7 + x1x4x7 + x2x3x7 + x4x5x6 + x3x5x6 + x1x5x6 + x3x4x6 + x2x4x6 + x2x3x6 + x1x3x6 + x1x2x6 + x3x4x5 + x1x3x5 + x1x2x4 + x1x2x3 + x6x7 + x4x7 + x3x7 + x2x7 + x1x7 + x3x6 + x4x5 + x3x5 + x1x5 + x3x4 + x3 + x2 + 1 x4x6x7 + x3x6x7 + x1x6x7 + x1x5x7 + x1x4x7 + x2x3x7 + x4x5x6 + x3x5x6 + x2x4x6 + x1x3x6 + x1x2x6 + x1x4x5 + x2x3x5 + x1x2x5 + x1x2x4 + x1x2x3 + x6x7 + x4x7 + x3x7 + x2x7 + x5x6 + x2x6 + x1x5 + x2x3 + x1x2 + x7 + x6 + x4 + x3 + x1 + 1x3x6x7 + x1x5x7 + x2x3x7 + x3x5x6 + x2x3x6 + x2x3x5 + x1x2x5 + x5x7 + x3x7 + x1x7 + x4x6 + x2x6 + x1x5 + x3x4 + x2x4 + x1x3 + x7 + x5 + x4 + x1 + 1x2x6x7 + x2x4x7 + x2x5x6 + x1x5x6 + x3x4x6 + x2x4x5 + x1x4x5 + x1x2x5 + x2x3x4 + x4x7 + x3x7 + x2x7 + x2x6 + x1x6 + x1x4 + x2x3 + x1x3 + x5 + x4 + x3 + x2 + 1 x1x5x7 + x3x4x7 + x2x3x7 + x1x5x6 + x3x4x6 + x2x3x6 + x3x4x5 + x1x4x5 + x2x3x5 + x1x3x5 + x1x2x5 + x5x7 + x5x7 + x4x7 + x3x7 + x4x6 + x1x6 + x4x5 + x2x5 + x1x2 +x1x5x6 + x3x4x6 + x2x3x6 + x1x4x5 + x2x3x4 + x5x7 + x2x7 + x5x6 + x1x6 + x4x5 + x1x5 + x1x3 + x1x2 + x5 + x3 + x2 + 1 x5x7 + x3x6 + x2x6 + x4x5 + x3x4 + x2x4 + x1x4 + x2x3 + x1x2 + x7 + x5 + x2 + 1 After elimination of x1, got 4 polynomials: x4x5x6x7 + x2x5x6x7 + x2x4x5x7 + x3x4x5x6 + x2x4x5x6 + x2x3x5x6 + x2x3x5x6 + x2x3x4x5 + x5x6x7 + x4x6x7 + x2x6x7 + x3x5x7 + x3x4x7 + x2x3x7 + x4x5x6 + x2x4x5x6 + x2x4x5x6 + x2x4x5x6 + x2x3x5x6 + x2x5x6x7 + x3x5x7 + x3x5x6 + x2x5x6 + x2x5x6 + x2x3x5x6 + x2x3x5x6 + x2x3x5x6 + x2x3x5x6 + x2x5x6x7 + x3x5x7 + x3x5x7 + x3x5x7 + x3x5x7 + x3x5x7 + x5x7 + x3x5x7 + x3x5x7 + x5x7 + x5x7 + x3x5x7 + x5x7 + x x2x7 + x5x6 + x4x6 + x3x6 + x4x5 + x3x5 + x2x5 + x7 + x6 + x5 + x3 + x2 + 1v7 ± 1 x2x5x6x7 + x2x4x6x7 + x2x3x6x7 + x3x4x5x7 + x2x4x5x7 + x2x4x5x6 + x2x3x5x6 + x2x3x4x6 + x2x3x4x5 + x2x6x7 + x3x5x7 + x3x4x7 + x2x4x6 + x3x4x5 + x2x4x5 + x2x3x5 + x2x3x4 + x4x7 + x2x4x6 + x2x3x6x7 + x2x4x6 + x2x3x6x7 + x2x4x6 + x2x3x6x7 + x2x6x7 + x2x6x7 + x2x4x6 + x2x3x6x7 + x2x4x5 + x2x3x6 + x2x3x6 + x2x3x6 + x2x3x6x7 + x2x6x7 + x2x6x7 + x2x4x5 + x2x3x6x7 + x2x6x7 + x2x6x7 + x2x6x7 + x2x4x6 + x2x3x6x7 + x2x6x7 + x2x6x7 + x2x4x5 + x2x3x6x7 + x2x6x7 + x2x7 + x2x6x7 + x2x7 + x2x6x7 + x2x7 + x2 x3x7 + x2x7 + x3x5 + x3x4 + x2x4 + x2x3 + x7 + x3 + x2 + 1x4x5x7 + x2x5x7 + x3x4x6 + x2x4x6 + x2x3x6 + x2x4x5 + x5x7 + x4x7 + x2x7 + x3x6 + x4x5 + x2x5 + x7 + x5 + x4 + x2 + 1

Increasing degree to max 5, General GRFY elimination of x_0, x_1

- Eliminating x_0 gives same 14 polynomials as over.
- Eliminating x_1 gives 16 polynomials.

Re-linearization and elimination of variables in Boolean equation systems | B. Greve, H.Raddum, Ø.Ytrehus 10/12

** Restricting degree to max. 4 **

After elimination of x0, got 14 polynomials:

x4x5x6x7 + x1x5x6x7 + x2x4x6x7 + x2x3x6x7 + x1x3x6x7 + x1x3x6x7 + x1x4x5x7 + x1x4x5x7 + x1x3x5x7 + x1x2x5x7 + x1x3x4x7 + x3x4x5x6 + x1x2x5x6 + x1x2x5x6 + x1x2x3x6 + x1x2x6 + x1x2x3x6 + x1x2x7 + x1x2x6 + x1x2x7 x3x4x5 + x1x2x3x4 + x5x6x7 + x3x6x7 + x4x5x7 + x2x5x7 + x1x5x7 + x3x4x7 + x2x3x7 + x1x3x7 + x4x5x6 + x2x5x6 + x1x5x6 + x2x4x6 + x2x3x6 + x1x3x6 + x3x4x5 + x1x3x5 + x1x3x5 + x1x3x5 + x1x2x 5 + x2x3x4 + x1x2x3 + x2x7 + x5x6 + x3x6 + x2x5 + x7 + x5 + x3 + x2 + 1 x3x5x6x7 + x1x5x6x7 + x2x4x6x7 + x2x3x6x7 + x1x2x6x7 + x3x4x5x7 + x1x2x4x7 + x1x2x4x7 + x1x2x3x7 + x2x4x5x6 + x1x3x4x6 + x1x3x4x6 + x1x3x4x5 + x1x2x4x5 + x1x2x3x5 + x3x5x7 + x3x4x5 + x1x2x4x7 + x1x2x7 + x1x7 + x1x2x7 + x1x7 + x1x x7 + x2x4x7 + x2x3x7 + x2x4x5 + x1x4x5 + x1x3x5 + x2x3x4 + x1x3x4 + x6x7 + x1x7 + x5x6 + x4x6 + x2x6 + x1x6 + x3x5 + x1x5 + x2x4 + x1x4 + x1x3 + x7 + x5 + x3 + x1 + 1 x2x5x6x7 + x1x5x6x7 + x1x4x6x7 + x2x4x5x7 + x1x4x5x7 + x1x2x5x7 + x2x3x4x7 + x2x3x5x6 + x1x3x5x6 + x1x3x4x6 + x1x2x4x6 + x2x3x4x5 + x1x2x3x4 + x5x6x7 + x2x6x7 + x2x6x7 + x2x5x7 + x2x5 x1x5x7 + x2x4x7 + x2x3x7 + x3x5x6 + x2x4x6 + x2x3x6 + x1x3x5 + x1x2x5 + x2x3x4 + x1x3x4 + x1x2x4 + x6x7 + x5x7 + x4x7 + x2x7 + x4x6 + x3x6 + x2x6 + x3x5 + x2x5 + x1x5 + x1x4 + x 7 + x6 + x5 + x3 + x2 + 1x1x5x6x7 + x2x3x6x7 + x1x3x5x7 + x1x3x5x7 + x1x2x3x7 + x2x3x5x6 + x1x2x5x6 + x1x3x4x6 + x1x3x4x5 + x1x2x3x4 + x2x6x7 + x2x3x7 + x1x3x7 + x1x2x7 + x2x5x6 + x1x5x6 + x1x3x4x6 + x 6 + x1x2x6 + x1x4x5 + x2x3x5 + x1x3x5 + x1x2x5 + x1x2x4 + x1x2x4 + x1x2x3 + x4x7 + x3x7 + x1x7 + x1x6 + x4x5 + x3x5 + x2x5 + x2x4 + x1x4 + x1x3 + x1x2 + x6 + x2 + x1 x3x4x6x7 + x1x4x6x7 + x1x2x6x7 + x3x4x5x7 + x1x4x5x7 + x2x3x5x7 + x1x2x5x7 + x1x2x4x7 + x1x2x4x7 + x1x2x3x7 + x3x4x5x6 + x1x2x4x6 + x1x2x4x6 + x1x2x4x6 + x1x3x4x5 + x1x2x4x6 + x1x2x6x6 + x1x2x6 + x1x2x6 + x1x2x6x6 + x1x2x6 + x1x2x6 + x1x2x6x6 + x1x2x6 + x2x3x4 + x4x6x7 + x1x6x7 + x2x5x7 + x3x4x7 + x2x4x7 + x1x4x7 + x2x3x7 + x4x5x6 + x2x4x6 + x2x3x6 + x1x3x6 + x1x2x6 + x2x4x5 + x2x3x4 + x1x2x3 + x6x7 + x5x7 + x5x6 + x3x6 + x2x6 + x2x4x6 + x2x3x6 + x1x3x6 + x1x2x6 + x2x4x7 + x1x4x7 + x5x7 + x5x6 + x3x6 + x2x4x6 + x2x3x6 + x1x3x6 + x1x2x6 + x2x4x7 + x1x2x3 + x6x7 + x5x7 + x5x6 + x3x6 + x2x4x6 + x2x3x6 + x1x3x6 + x1x2x6 + x2x4x6 + x2x4x6 + x2x4x6 + x2x4x6 + x2x4x6 + x2x4x7 + x1x4x7 + x5x6 + x2x4x6 + x2x4x6 + x2x3x6 + x1x3x6 + x1x2x6 + x2x4x6 + x2 x1x6 + x3x5 + x2x4 + x7 + x5 + x3 + 1x2x4x6x7 + x1x3x6x7 + x1x2x6x7 + x1x3x4x7 + x1x2x4x7 + x3x4x5x6 + x2x4x5x6 + x1x4x5x6 + x2x3x5x6 + x1x2x5x6 + x1x2x5x6 + x1x2x3x6 + x2x3x4x5 + x5x6x7 + x4x6x7 + x3x6x7 + x2x6x7 + x2x6x7 + x2x6x7 + x2x6x7 + x3x6x7 + x2x6x7 + x3x6x7 + x3x7x7 + x3x7x7 + x3x7x7 + x3x7x7 + x3x7 x1x6x7 + x4x5x7 + x3x4x7 + x1x4x7 + x1x3x7 + x1x2x7 + x4x5x6 + x2x5x6 + x1x5x6 + x3x4x6 + x2x4x6 + x1x2x6 + x2x4x5 + x1x4x5 + x2x3x5 + x1x3x5 + x1x2x5 + x2x3x4 + x1x3x7 + x1x2x4 + x6x7 + x5x7 + x4x7 + x3x7 + x2x7 + x1x7 + x5x6 + x4x6 + x3x6 + x1x6 + x4x5 + x2x5 + x1x4 + x1x3 + x1x2 + x7 + x5 + x3 + x2 + x1 + 1x1x4x6x7 + x2x3x6x7 + x1x3x6x7 + x1x2x6x7 + x3x4x5x7 + x1x4x5x7 + x2x3x5x7 + x1x3x5x7 + x1x3x4x7 + x1x2x4x7 + x1x4x5x6 + x2x3x5x6 + x1x2x5x6 + x1x2x4x6 + x1x2x3x6 + x2x3x4x5 + x1 x3x4x5 + x1x2x3x4 + x5x6x7 + x2x6x7 + x4x5x7 + x2x5x7 + x3x4x7 + x1x3x7 + x1x2x7 + x3x5x6 + x1x5x6 + x3x4x6 + x2x3x6 + x1x3x6 + x2x4x5 + x1x3x5 + x1x2x5 + x1x2x5 + x1x2x4 + x1x2x3 + x6x7 + x3x7 + x1x6 + x4x5 + x3x5 + x3x4 + x2x4 + x2x3 + x1x2 + x4x5x6x7 + x4x6x7 + x2x6x7 + x1x6x7 + x1x5x7 + x1x5x7 + x3x4x7 + x2x4x7 + x1x4x7 + x2x3x7 + x4x5x6 + x3x5x6 + x1x5x6 + x3x4x6 + x2x4x6 + x2x3x6 + x1x3x6 + x1x2x6 + x3x4x5 + x1x3x5 + x1x2x4 + x1x2x3 + x6x7 + x4x7 + x3x7 + x2x7 + x1x7 + x3x6 + x4x5 + x3x5 + x1x5 + x3x4 + x3 + x2 + 1 x4x6x7 + x3x6x7 + x1x6x7 + x1x5x7 + x1x4x7 + x2x3x7 + x4x5x6 + x3x5x6 + x2x4x6 + x1x3x6 + x1x2x6 + x1x4x5 + x2x3x5 + x1x2x5 + x1x2x4 + x1x2x3 + x6x7 + x4x7 + x3x7 + x2x7 + x5x6 + x2x6 + x1x5 + x2x3 + x1x2 + x7 + x6 + x4 + x3 + x1 + 1x3x6x7 + x1x5x7 + x2x3x7 + x3x5x6 + x2x3x6 + x2x3x5 + x1x2x5 + x5x7 + x3x7 + x1x7 + x4x6 + x2x6 + x1x5 + x3x4 + x2x4 + x1x3 + x7 + x5 + x4 + x1 + 1x2x6x7 + x2x4x7 + x2x5x6 + x1x5x6 + x3x4x6 + x2x4x5 + x1x4x5 + x1x2x5 + x2x3x4 + x4x7 + x3x7 + x2x7 + x2x6 + x1x6 + x1x4 + x2x3 + x1x3 + x5 + x4 + x3 + x2 + 1 x1x5x7 + x3x4x7 + x2x3x7 + x1x5x6 + x3x4x6 + x2x3x6 + x3x4x5 + x1x4x5 + x2x3x5 + x1x3x5 + x1x2x5 + x5x7 + x5x7 + x4x7 + x3x7 + x4x6 + x1x6 + x4x5 + x2x5 + x1x2 +x1x5x6 + x3x4x6 + x2x3x6 + x1x4x5 + x2x3x4 + x5x7 + x2x7 + x5x6 + x1x6 + x4x5 + x1x5 + x1x3 + x1x2 + x5 + x3 + x2 + 1 x5x7 + x3x6 + x2x6 + x4x5 + x3x4 + x2x4 + x1x4 + x2x3 + x1x2 + x7 + x5 + x2 + 1 After elimination of x1, got 4 polynomials: x4x5x6x7 + x2x5x6x7 + x2x4x5x7 + x3x4x5x6 + x2x4x5x6 + x2x3x5x6 + x2x3x5x6 + x2x3x4x5 + x5x6x7 + x4x6x7 + x2x6x7 + x3x5x7 + x3x4x7 + x2x3x7 + x4x5x6 + x2x4x5x6 + x2x4x5x6 + x2x4x5x6 + x2x3x5x6 + x2x5x6x7 + x3x5x7 + x3x5x6 + x2x5x6 + x2x5x6 + x2x3x5x6 + x2x3x5x6 + x2x3x5x6 + x2x3x5x6 + x2x5x6x7 + x3x5x7 + x3x5x7 + x3x5x7 + x3x5x7 + x3x5x7 + x5x7 + x3x5x7 + x3x5x7 + x5x7 + x5x7 + x3x5x7 + x5x7 + x x2x7 + x5x6 + x4x6 + x3x6 + x4x5 + x3x5 + x2x5 + x7 + x6 + x5 + x3 + x2 + 1v7 ± 1 x2x5x6x7 + x2x4x6x7 + x2x3x6x7 + x3x4x5x7 + x2x4x5x7 + x2x4x5x6 + x2x3x5x6 + x2x3x4x6 + x2x3x4x5 + x2x6x7 + x3x5x7 + x3x4x7 + x2x4x6 + x3x4x5 + x2x4x5 + x2x3x5 + x2x3x4 + x4x7 + x2x4x6 + x2x3x6x7 + x2x4x6 + x2x3x6x7 + x2x4x6 + x2x3x6x7 + x2x6x7 + x2x6x7 + x2x4x6 + x2x3x6x7 + x2x4x5 + x2x3x6 + x2x3x6 + x2x3x6 + x2x3x6x7 + x2x6x7 + x2x6x7 + x2x4x5 + x2x3x6x7 + x2x6x7 + x2x6x7 + x2x6x7 + x2x4x6 + x2x3x6x7 + x2x6x7 + x2x6x7 + x2x4x5 + x2x3x6x7 + x2x6x7 + x2x7 + x2x6x7 + x2x7 + x2x6x7 + x2x7 + x2 x3x7 + x2x7 + x3x5 + x3x4 + x2x4 + x2x3 + x7 + x3 + x2 + 1x4x5x7 + x2x5x7 + x3x4x6 + x2x4x6 + x2x3x6 + x2x4x5 + x5x7 + x4x7 + x2x7 + x3x6 + x4x5 + x2x5 + x7 + x5 + x4 + x2 + 1

Increasing degree to max 5, General GRFY elimination of x_0, x_1

- Eliminating x_0 gives same 14 polynomials as over.
- Eliminating x₁ gives 16 polynomials.



** Restricting degree to max. 4 **

After elimination of x0, got 14 polynomials:

x4x56x7 + x1x56x7 + x2x46x7 + x2x36x67 + x1x3x6x7 + x1x4x5x7 + x1x4x5x7 + x1x2x5x7 + x1x2x5x7 + x1x2x5x7 + x1x4x5x6 + x2x3x6x6 + x1x25x6 + x1x25x6 + x1x25x7 + x1x4x7 + x1x4x5x7 + x1x4x7 + x1
$5 + x^2x^3x^4 + x^1x^2x^3 + x^2x^7 + x^5x^6 + x^3x^6 + x^2x^5 + x^7 + x^5 + x^3 + x^2 + 1$
x3x5x5x7 + x1x5x5x7 + x2x4x5x7 + x2x3x5x7 + x1x2x5x7 + x1x2x5x7 + x1x3x4x7 + x1x2x4x7 + x1x2x4x7 + x1x2x4x7 + x1x4x5x6 + x1x4x5x6 + x1x3x4x6 + x2x3x4x5 + x1x3x4x5 + x1x2x3x5 + x1x2x3x5 + x1x2x3x5 + x1x2x4x5 + x1x2x3x5 + x1x2x3x5 + x1x2x3x5 + x1x2x4x5 + x1x2x4x5 + x1x2x3x5 + x1x2x3x5 + x1x2x4x5 + x1x2x3x5 + x1x2x3x5 + x1x2x4x5 + x1x2x4x5 + x1x2x4x5 + x1x2x3x5 + x1x2x4x5 + x1x2x4x5 + x1x2x3x5 + x1x2x4x5 + x1x2x5
$x^{7} + x^{2}x^{4}x^{7} + x^{2}x^{4}x^{5} + x^{1}x^{4}x^{5} + x^{1}x^{3}x^{5} + x^{2}x^{3}x^{4} + x^{1}x^{3}x^{4} + x^{6}x^{7} + x^{1}x^{6} + x^{6}x^{6} + x^{4}x^{6} + x^{2}x^{6} + x^{1}x^{6} + x^{3}x^{5} + x^{1}x^{5} + x^{1}x^{4} + x^{1}x^{3} + x^{7} + x^{5} + x^{1} + 1$
2425567 + x1x3567 + x1x4567 + x2x45x7 + x1x4557 + x1x257 + x1x2x57 + x1x2x5x6 + x1x35x6 + x1x3x4x6 + x1x2x4x6 +
$x_{24,500}$, $t_{14,2300}$, $t_{14,2430}$, $t_{24,247}$, $x_{25,56}$, $t_{22,46}$, $x_{24,56}$, $t_{24,25,200}$, $t_{14,25,200}$, $t_{14,25,$
7 + x6 + x5 + x3 + x2 + 1
x1x5x6x7 + x2x3x6x7 + x1x3x6x7 + x1x3x5x7 + x1x2x3x7 + x2x3x5x6 + x1x2x5x6 + x1x3x4x6 + x1x3x4x5 + x1x2x3x4 + x2x6x7 + x2x3x7 + x1x3x7 + x1x2x7 + x2x5x6 + x1x5x6 + x1x5x6 + x1x3x4x6 + x1x3x4x5 + x1x2x6x7 + x2x6x7 + x2x5x7 + x2x5x6 + x1x5x6 + x1x5x6 + x1x3x4x6 + x1x3x4x5 + x1x2x6x7 + x2x6x7 + x1x3x7 + x1x2x7 + x2x5x6 + x1x5x6 + x1x5x6 + x1x3x4x6 + x1x3x4x5 + x1x5x6 + x1x
6 + x1x2x6 + x1x4x5 + x2x3x5 + x1x3x5 + x1x3x5 + x1x2x5 + x1x3x4 + x1x2x4 + x1x2x3 + x4x7 + x3x7 + x2x7 + x1x7 + x1x6 + x4x5 + x3x5 + x2x4 + x1x4 + x1x3 + x1x2 + x6 + x2 + x1
x3x4x6x7 + x1x4x6x7 + x1x2x6x7 + x1x2x6x7 + x1x4x5x7 + x1x3x4x5x7 + x1x2x5x7 + x1x2x4x7 + x1x2x4x7 + x1x2x4x7 + x1x2x4x5 + x1x4x5x6 + x1x2x4x6 + x1x2x6 + x1x2x4x6 + x1x2x6x6 + x1x2x6x6 + x1x2x6x6 + x1x2x6 + x1x2x6x6 + x1x2x6 + x1x2x6x6 + x1x2x6x6 + x1x2x6x6 + x1x2
x2x3x4 + x4x6x7 + x1x6x7 + x2x5x7 + x3x4x7 + x2x4x7 + x1x4x7 + x2x3x7 + x4x5x6 + x2x4x6 + x2x3x6 + x1x3x6 + x1x2x6 + x2x4x5 + x2x3x4 + x1x2x3 + x6x7 + x5x7 + x5x6 + x3x6 + x2x6 + x2x6 + x2x3x6 + x1x2x6 + x2x6x7 + x1x2x7 + x1x2x7 + x5x7 + x5
x1x6 + x3x5 + x2x4 + x7 + x5 + x3 + 1
x2x4x6x7 + x1x3x6x7 + x1x2x6x7 + x1x3x4x7 + x1x2x4x7 + x1x2x4x7 + x3x4x5x6 + x2x4x5x6 + x1x4x5x6 + x2x3x5x6 + x1x2x5x6 + x1x2x5x6 + x1x2x3x6 + x2x3x4x5 + x5x6x7 + x4x6x7 + x3x6x7 + x2x6x7 + x2x6x7 + x1x2x5x6 + x1x2x5x5x6 + x1x2x5x6 + x1x2x5x6 + x1x2x5x6 + x1x2x5x6 + x1x2x5x6 + x1x2x5x6
x1x5x7 + x4x5x7 + x3x4x7 + x2x4x7 + x1x4x7 + x1x3x7 + x1x2x7 + x4x5x6 + x2x5x6 + x1x5x6 + x3x4x6 + x2x4x6 + x1x2x6 + x2x4x5 + x1x4x5 + x2x3x5 + x1x3x7 + x1x2x7 + x4x5x6 + x1x3x7
+ x1x2x4 + x6x7 + x5x7 + x4x7 + x3x7 + x2x7 + x1x7 + x5x6 + x4x6 + x3x6 + x1x6 + x4x5 + x2x5 + x1x4 + x1x3 + x1x2 + x7 + x5 + x3 + x2 + x1 + 1
x1x4x6x7 + x2x3x6x7 + x1x3x6x7 + x1x2x6x7 + x3x4x5x7 + x1x4x5x7 + x2x3x5x7 + x1x3x5x7 + x1x3x4x7 + x1x2x4x7 + x1x4x5x6 + x1x2x5x6 + x1x2x5x6 + x1x2x4x6 + x1x2x3x6 + x1x2x6 + x1x2x6 + x1x2x6 + x1x2x3x6 + x1x2x6 +
x3x4x5 + x1x2x3x4 + x5x6x7 + x2x6x7 + x4x5x7 + x2x5x7 + x3x4x7 + x1x3x7 + x1x2x7 + x3x5x6 + x1x5x6 + x3x4x6 + x2x3x6 + x1x3x6 + x2x4x5 + x1x3x5 + x1x2x4 + x1x2x3 + x6x7
+ x3x7 + x1x6 + x4x5 + x3x5 + x3x4 + x2x4 + x2x3 + x1x2 + x4
x5x6x7 + x4x6x7 + x2x6x7 + x1x6x7 + x4x5x7 + x1x5x7 + x1x5x7 + x2x4x7 + x1x4x7 + x2x3x7 + x4x5x6 + x3x5x6 + x1x5x6 + x3x4x6 + x2x4x6 + x2x3x6 + x1x3x6 + x1x2x6 + x3x4x5 + x1x3x5
+ x1x2x4 + x1x2x3 + x6x7 + x4x7 + x3x7 + x2x7 + x1x7 + x3x6 + x4x5 + x3x5 + x1x5 + x3x4 + x3 + x2 + 1
x4x6x7 + x3x6x7 + x1x6x7 + x1x5x7 + x1x4x7 + x2x3x7 + x4x5x6 + x3x5x6 + x2x4x6 + x1x3x6 + x1x2x6 + x1x4x5 + x2x3x5 + x1x2x5 + x1x2x4 + x1x2x3 + x6x7 + x4x7 + x3x7 + x2x7 + x5x6 +
$x^{2x6} + x^{1x5} + x^{2x3} + x^{1x2} + x^{7} + x^{6} + x^{4} + x^{3} + x^{1} + 1$
x3x6x7 + x1x5x7 + x2x3x7 + x3x5x6 + x2x3x6 + x2x3x5 + x1x2x5 + x5x7 + x3x7 + x1x7 + x4x6 + x2x6 + x1x5 + x3x4 + x2x4 + x1x3 + x7 + x5 + x4 + x1 + 1
$x^{2}x^{6}x^{7} + x^{2}x^{4}x^{7} + x^{2}x^{5}x^{6} + x^{1}x^{5}x^{6} + x^{3}x^{4}x^{6} + x^{2}x^{4}x^{5} + x^{1}x^{2}x^{5} + x^{2}x^{3}x^{4} + x^{4}x^{7} + x^{3}x^{7} + x^{2}x^{7} + x^{2}x^{6} + x^{1}x^{6} + x^{1}x^{4} + x^{2}x^{3} + x^{1}x^{3} + x^{5} + x^{4} + x^{3} + x^{2} + 1$
x1x5x7 + x3x4x7 + x2x3x7 + x1x5x6 + x3x4x6 + x2x3x6 + x3x4x5 + x1x4x5 + x2x3x5 + x1x3x5 + x1x3x5 + x1x2x5 + x6x7 + x5x7 + x4x7 + x3x7 + x4x6 + x1x6 + x1x6 + x4x5 + x2x5 + x1x4 + x1x2 + x1
x1x5x6 + x3x4x6 + x2x3x6 + x1x4x5 + x2x3x4 + x5x7 + x2x7 + x5x6 + x1x6 + x4x5 + x1x5 + x1x3 + x1x2 + x5 + x3 + x2 + 1
x5x7 + x3x6 + x2x6 + x4x5 + x3x4 + x2x4 + x1x4 + x2x3 + x1x2 + x7 + x5 + x2 + 1
After elimination of x1, got 4 polynomials:
x4x5x6x7 + x2x5x6x7 + x2x4x5x7 + x3x4x5x6 + x2x4x5x6 + x2x3x5x6 + x2x3x5x6 + x2x3x4x5 + x5x6x7 + x4x6x7 + x2x6x7 + x3x4x7 + x2x3x4x7 + x4x5x6 + x2x5x6 + x2x4x5 + x6x7 + x5x7 + x3x7 + x4x5x7 + x3x4x7 + x2x5x6 + x2x5x6 + x2x5x6 + x2x4x5 + x5x7 + x3x7 + x5x7 + x3x5x7 + x3x4x7 + x5x7 +
x2x7 + x5x6 + x4x6 + x3x6 + x4x5 + x3x5 + x2x5 + x7 + x6 + x5 + x3 + x2 + 1
33X55657 + x3x4557 + x3x457 + x3x457 + x3x477 + x3x457 + x3x477 + x3x457 + x3x475 + x3x55 +
A2 + 1 X2X5X6X7 + X2X4X6X7 + X2X3X6X7 + X3X4X5X7 + X2X4X5X7 + X2X4X5X6 + X2X3X5X6 + X2X3X4X6 + X2X3X4X5 + X2X6X7 + X3X5X7 + X3X4X7 + X2X4X6 + X3X4X5 + X2X3X5 + X2X3X4 + X4X7 +
AZAJONI T AZANONI T AZANONI T AZANAJNI T AZANAJNI T AZANAJNI T AZANAJNI T AZANAJNI T AZANANI T AJANANI T AZANAJ T AZANAJNI TAZANI T AZANAJNI T AZANAJNI T AZANAJNI T AZANAJNI T AZANAJNI TAZANI TAZANAJNI TAZANI TAZANAJNI TAZANAJNI TAZANI TAZANAJNI TAZANAJNI TAZANI TAZANAJNI TA
$x_{2}x_{1}$ + $x_{2}x_{1}$ + $x_{2}x_{3}$ + $x_{3}x_{3}$ + $x_{4}x_{3}$ + $x_{2}x_{3}$ + $x_{2}x_{3}$ + $x_{3}x_{5}$ + $x_{3}x_{5}$ + $x_{3}x_{5}$ + $x_{3}x_{5}$ + $x_{4}x_{5}$ + $x_{2}x_{5}$ + $x_{4}x_{5}$ + x_{2} + x_{2} + 1
X4XXX T X2XX T X X2XX T X X2XX T X XXX X X X XXX T X XXX X

Increasing degree to max 5, General GRFY elimination of x_0, x_1

- Eliminating x_0 gives same 14 polynomials as over.
- Eliminating x_1 gives 16 polynomials.

** Restricting degree to max. 4 **

After elimination of x0, got 14 polynomials:

x4x5x6x7 + x1x5x6x7 + x2x4x6x7 + x2x3x6x7 + x1x3x6x7 + x3x4x5x7 + x1x4x5x7 + x2x3x5x7 + x1x3x5x7 + x1x2x5x7 + x1x3x4x7 + x3x4x5x6 + x2x3x5x6 + x1x2x5x6 + x1x2x3x6 + x1x2x6 + x1x2x3x6 + x1x2x6
x3x4x5 + x1x2x3x4 + x5x6x7 + x3x6x7 + x4x5x7 + x1x5x7 + x1x5x7 + x1x5x7 + x1x3x7 + x1x3x7 + x4x5x6 + x2x5x6 + x1x5x6 + x2x3x6 + x1x3x6 + x1x3x6 + x2x3x6 + x1x3x5 + x1x5x5 + x
3 + 2243/4 + 11425/6 7 + 224/6/6 7 + 224/6/6 7 + 224/6/6 7 + 224/6/6 7 + 224/6/6 7 + 224/6/6 7 + 224/6/6 7 + 224/6/6 + 21/2/6 + 21/2/6 + 21/2/6/6 + 21/2/6 +
$x_{2} + x_{2} + x_{3} + x_{4} + x_{4} + x_{4} + x_{1} + x_{1} + x_{2} + x_{4} + x_{4$
1/2 + $1/2$ + $2/2$ + $1/2$ + $2/2$ + $1/2$
$\lambda_{2453004}$, $\lambda_{1123004}$, λ_{124307} , λ_{124374} , λ_{124374} , $\lambda_{1242374}$, $\lambda_{1242374}$, λ_{242374} , λ_{125374} , λ_{124277} , λ_{2237} , λ_{23756} , λ_{2256} , λ_{2375} , λ_{2477} , λ_{2377} , λ_{2376} , λ_{23756} , λ_{2375} , $\lambda_$
XIXXX T XXXXX T XXXX T XXXXX T XXXXXX
/ * * * * * * * * * * * * * * * * * * *
$\frac{1}{1} + \frac{1}{1} + \frac{1}$
0 + 1.1220 + 1.12430 + 1.12430 + 1.1243 + 1.1243 + 1.12430 + 1.1
ASAMADA' + AIAKADA' + AIAKADA' + ASAMADA' + AIAKADA' + AIAKADA' + AIAKADA' + AIAKAMA' + AIAKADA' + AIAKAD + AIAKADA + AIAKAD + AIAKADA + AIAKAD + A
x2x4x5x7 + x1x2x5x7 + x1x2x6x7 + x1x2x4x7 + x1x2x4x7 + x1x2x4x7 + x3x4x5x6 + x2x4x5x6 + x1x4x5x6 + x2x3x5x6 + x1x2x5x6 + x1x2x3x6 + x1x2x3x6 + x2x3x4x5 + x5x6x7 + x4x6x7 + x3x6x7 + x2x6x7 + x2x6x7 + x1x2x5x6 +
x1x6x7 + x4x5x7 + x3x4x7 + x2x4x7 + x1x4x7 + x1x3x7 + x1x2x7 + x4x5x6 + x2x5x6 + x1x5x6 + x3x4x6 + x2x4x6 + x1x2x6 + x2x4x5 + x1x4x5 + x1x3x5 + x1x3x5 + x1x2x5 + x1x3x4
+ x1x2x4 + x6x7 + x5x7 + x4x7 + x3x7 + x2x7 + x1x7 + x5x6 + x4x6 + x3x6 + x1x6 + x4x5 + x2x5 + x1x4 + x1x3 + x1x2 + x7 + x5 + x3 + x2 + x1 + 1
x1x4x6x7 + x2x3x6x7 + x1x3x6x7 + x1x2x6x7 + x3x4x5x7 + x1x4x5x7 + x2x3x5x7 + x1x3x5x7 + x1x3x5x7 + x1x3x4x7 + x1x4x5x6 + x2x3x5x6 + x1x2x5x6 + x1x2x4x6 + x1x2x6 + x1x2x
x3x4x5 + x1x2x3x4 + x5x6x7 + x2x6x7 + x4x5x7 + x2x5x7 + x3x4x7 + x1x3x7 + x1x2x7 + x3x5x6 + x1x5x6 + x1x5x6 + x2x3x6 + x1x3x6 + x2x4x5 + x1x3x5 + x1x2x5 + x1x2x5 + x1x2x3 + x6x7
+ x3x7 + x1x6 + x4x5 + x3x5 + x3x4 + x2x4 + x2x3 + x1x2 + x4
x5x6x7 + x4x6x7 + x2x6x7 + x1x6x7 + x1x6x7 + x1x5x7 + x1x5x7 + x1x5x7 + x2x4x7 + x1x4x7 + x2x3x7 + x4x5x6 + x3x5x6 + x1x5x6 + x1x5x6 + x2x4x6 + x2x3x6 + x1x3x6 + x1x3x6 + x1x2x6 + x3x4x5 + x1x3x5
+ x1x2x4 + x1x2x3 + x6x7 + x4x7 + x3x7 + x2x7 + x1x7 + x3x6 + x4x5 + x3x5 + x1x5 + x3x4 + x3 + x2 + 1
x4x6x7 + x3x6x7 + x1x6x7 + x1x5x7 + x1x5x7 + x1x4x7 + x2x3x7 + x4x5x6 + x3x5x6 + x2x4x6 + x1x3x6 + x1x2x6 + x1x4x5 + x2x3x5 + x1x2x5 + x1x2x4 + x1x2x3 + x6x7 + x4x7 + x3x7 + x2x7 + x5x6 + x1x2x6 + x1x2x6 + x1x4x5 + x1x2x6 + x1x2
x2x6 + x1x5 + x2x3 + x1x2 + x7 + x6 + x4 + x3 + x1 + 1
x3x6x7 + x1x5x7 + x2x3x7 + x3x5x6 + x2x3x6 + x2x3x5 + x1x2x5 + x1x2x5 + x3x7 + x1x7 + x4x6 + x2x6 + x1x5 + x3x4 + x2x4 + x1x3 + x7 + x5 + x4 + x1 + 1
x2x6x7 + x2x4x7 + x2x5x6 + x1x5x6 + x3x4x6 + x2x4x5 + x1x4x5 + x1x2x5 + x2x3x4 + x4x7 + x3x7 + x2x7 + x2x6 + x1x6 + x1x4 + x2x3 + x1x3 + x5 + x4 + x3 + x2 + 1
x1x5x7 + x3x4x7 + x2x3x7 + x1x5x6 + x3x4x6 + x2x3x6 + x3x4x5 + x1x4x5 + x1x4x5 + x1x3x5 + x1x2x5 + x6x7 + x5x7 + x4x7 + x3x7 + x4x6 + x1x6 + x4x5 + x2x5 + x1x4 + x1x2 + x1
x1x5x6 + x3x4x6 + x2x3x6 + x1x4x5 + x2x3x4 + x5x7 + x2x7 + x5x6 + x1x6 + x4x5 + x1x5 + x1x3 + x1x2 + x5 + x3 + x2 + 1
x5x7 + x3x6 + x2x6 + x4x5 + x3x4 + x2x4 + x1x4 + x2x3 + x1x2 + x7 + x5 + x2 + 1
After elimination of x1, got 4 polynomials:
x4x5x6x7 + x2x5x6x7 + x2x4x5x7 + x3x4x5x6 + x2x4x5x6 + x2x3x5x6 + x2x3x5x6 + x2x3x5x6 + x2x4x5 + x6x67 + x2x5x7 + x3x5x7 + x3x4x7 + x2x3x7 + x4x5x6 + x2x5x6 + x2x4x5 + x6x7 + x5x7 + x3x7 + x2x7 + x5x7 + x3x7 + x2x6 + x4x5 + x6x7 + x3x5 + x6x7 + x5 + x1 + x2 + 1
xx/ * x5xb * x*xb + x3xb * x4x5 + x4x5 + x2x5 + x2x + x7 + x0 + x5 + x3 * x2 + x1 x3x5xb67 + x3x4x5x7 + x3x4x5x7 + x3x5xb67 + x3x5x7 + x3x5x7 + x3x5x6 + x2x4x6 + x3x4x5 + x2x4x5 + x2x3x4 + x5x7 + x2x7 + x4x5 + x3x5 + x2x5 + x3x4 + x2x3 + x7 + x6 + x5 + x2x4x5 + x2x5 + x2x5 + x2x5 + x2x5 + x2x5 + x2x4x5 + x2x5 + x2
+ CX + 0X + 1X + CX2X + PXCX + CX2X + CX2X + CX4XX + 1X2X + 1X2X + CX4XXX + CX4XXX + CX4XXX + 0X4XXX + 0X4XXX + 1X2XXX +
^2 + 1 x2x5x6x7 + x2x4x6x7 + x2x3x6x7 + x3x4x5x7 + x2x4x5x7 + x2x4x5x6 + x2x3x5x6 + x2x3x4x6 + x2x3x4x5 + x2x6x7 + x3x5x7 + x3x4x7 + x2x4x6 + x3x4x5 + x2x4x5 + x2x3x5 + x2x3x4 + x4x7 +
AZEJADAY + JZENYADAY + AZENYADAY + AZENYAJA + AZENYAJA + ZZENYADA + ZZENYADA + ZZENYADA + ZZENYAD + ZZENY
$x_{25}(x_{1}^{-1}, x_{25}(x_{1}^{-1}, x_{3}(x_{1}^{-1}, x_{3}(x_{1}^{-1}, x_{3}^{-1}, x_$

Increasing degree to max 5, General GRFY elimination of x_0, x_1

- Eliminating x_0 gives same 14 polynomials as over.
- Eliminating x_1 gives 16 polynomials.

First elimination ideal, degree ≤ 3

- Number of resultants $\binom{m}{2}$, of coefficient constraints m
- Number of polynomials produced of elimination: $\binom{m}{2} + m$
- Number of monomials of degree 3: $\sum_{i=1}^{3} \binom{n}{i}$
- $\binom{m}{2} + m < \sum_{i=1}^{3} \binom{n}{i} \rightarrow \text{no re-linearization.}$

- Number of resultants $\binom{m}{2}+m$, of coefficient constraints $\binom{m}{2}+m$.
- Number of polynomials produced of elimination: $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m$.
- Number of monomials of degree 5: $\sum_{i=1}^{5} \binom{n}{i}$
- $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m > \sum_{i=1}^{5} \binom{n}{i}?$

First elimination ideal, degree ≤ 3

- Number of resultants $\binom{m}{2}$, of coefficient constraints m
- Number of polynomials produced of elimination: $\binom{m}{2} + m$
- Number of monomials of degree 3: $\sum_{i=1}^{3} \binom{n}{i}$
- $\binom{m}{2} + m < \sum_{i=1}^{3} \binom{n}{i} \rightarrow \text{no re-linearization.}$

- Number of resultants $\binom{m}{2}+m$, of coefficient constraints $\binom{m}{2}+m$.
- Number of polynomials produced of elimination: $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m$.
- Number of monomials of degree 5: $\sum_{i=1}^{5} \binom{n}{i}$
- $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m > \sum_{i=1}^{5} \binom{n}{i}?$

First elimination ideal, degree ≤ 3

- Number of resultants $\binom{m}{2}$, of coefficient constraints m
- Number of polynomials produced of elimination: $\binom{m}{2} + m$
- Number of monomials of degree 3: $\sum_{i=1}^{3} \binom{n}{i}$
- $\binom{m}{2} + m < \sum_{i=1}^{3} \binom{n}{i} \rightarrow \text{no re-linearization.}$

- Number of resultants $\binom{m}{2}+m$, of coefficient constraints $\binom{m}{2}+m$.
- Number of polynomials produced of elimination: $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m$.
- Number of monomials of degree 5: $\sum_{i=1}^{5} \binom{n}{i}$
- $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m > \sum_{i=1}^{5} \binom{n}{i}?$

First elimination ideal, degree ≤ 3

- Number of resultants $\binom{m}{2}$, of coefficient constraints m
- Number of polynomials produced of elimination: $\binom{m}{2} + m$
- Number of monomials of degree 3: $\sum_{i=1}^{3} \binom{n}{i}$

• $\binom{m}{2} + m < \sum_{i=1}^{3} \binom{n}{i} \to \text{no re-linearization.}$

- Number of resultants $\binom{m}{2}+m$, of coefficient constraints $\binom{m}{2}+m$.
- Number of polynomials produced of elimination: $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m$.
- Number of monomials of degree 5: $\sum_{i=1}^{5} \binom{n}{i}$
- $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m > \sum_{i=1}^{5} \binom{n}{i}?$

First elimination ideal, degree ≤ 3

- Number of resultants $\binom{m}{2}$, of coefficient constraints m
- Number of polynomials produced of elimination: $\binom{m}{2} + m$
- Number of monomials of degree 3: $\sum_{i=1}^{3} \binom{n}{i}$
- $\binom{m}{2} + m < \sum_{i=1}^{3} \binom{n}{i} \rightarrow \text{no re-linearization.}$

- Number of resultants $\binom{\binom{m}{2}+m}{2}$, of coefficient constraints $\binom{m}{2}+m$.
- Number of polynomials produced of elimination: $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m$.
- Number of monomials of degree 5: $\sum_{i=1}^{5} \binom{n}{i}$
- $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m > \sum_{i=1}^{5} \binom{n}{i}?$

First elimination ideal, degree ≤ 3

- Number of resultants $\binom{m}{2}$, of coefficient constraints m
- Number of polynomials produced of elimination: $\binom{m}{2} + m$
- Number of monomials of degree 3: $\sum_{i=1}^{3} \binom{n}{i}$
- $\binom{m}{2} + m < \sum_{i=1}^{3} \binom{n}{i} \rightarrow \text{no re-linearization.}$

- Number of resultants $\binom{\binom{m}{2}+m}{2}$, of coefficient constraints $\binom{m}{2}+m$.
- Number of polynomials produced of elimination: $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m$
- Number of monomials of degree 5: $\sum_{i=1}^{5} \binom{n}{i}$
- $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m > \sum_{i=1}^{5} \binom{n}{i}?$

First elimination ideal, degree ≤ 3

- Number of resultants $\binom{m}{2}$, of coefficient constraints m
- Number of polynomials produced of elimination: $\binom{m}{2} + m$
- Number of monomials of degree 3: $\sum_{i=1}^{3} \binom{n}{i}$
- $\binom{m}{2} + m < \sum_{i=1}^{3} \binom{n}{i} \rightarrow \text{no re-linearization.}$

- Number of resultants $\binom{\binom{m}{2}+m}{2}$, of coefficient constraints $\binom{m}{2}+m$.
- Number of polynomials produced of elimination: $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m$.
- Number of monomials of degree 5: $\sum_{i=1}^{5} \binom{n}{i}$
- $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m > \sum_{i=1}^{5} \binom{n}{i}?$

First elimination ideal, degree ≤ 3

- Number of resultants $\binom{m}{2}$, of coefficient constraints m
- Number of polynomials produced of elimination: $\binom{m}{2} + m$
- Number of monomials of degree 3: $\sum_{i=1}^{3} \binom{n}{i}$
- $\binom{m}{2} + m < \sum_{i=1}^{3} \binom{n}{i} \rightarrow \text{no re-linearization.}$

- Number of resultants $\binom{m}{2}+m$, of coefficient constraints $\binom{m}{2}+m$.
- Number of polynomials produced of elimination: $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m$.
- Number of monomials of degree 5: $\sum_{i=1}^{5} \binom{n}{i}$

First elimination ideal, degree ≤ 3

- Number of resultants $\binom{m}{2}$, of coefficient constraints m
- Number of polynomials produced of elimination: $\binom{m}{2} + m$
- Number of monomials of degree 3: $\sum_{i=1}^{3} \binom{n}{i}$
- $\binom{m}{2} + m < \sum_{i=1}^{3} \binom{n}{i} \rightarrow \text{no re-linearization.}$

Second elimination ideal, degree ≤ 5

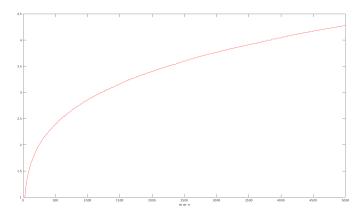
- Number of resultants $\binom{m}{2}+m$, of coefficient constraints $\binom{m}{2}+m$.
- Number of polynomials produced of elimination: $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m$.
- Number of monomials of degree 5: $\sum_{i=1}^{5} \binom{n}{i}$

•
$$\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m > \sum_{i=1}^{5} \binom{n}{i}?$$

11/12

Re-linearization curve

3-5 version2.png



When m = n this holds true for $1 \le n \le 25$

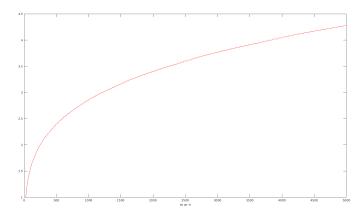
Re-linearization and elimination of variables in Boolean equation systems

B. Greve, H.Raddum, Ø.Ytrehus

12 / 12

Re-linearization curve

3-5 version2.png



When m = n this holds true for $1 \le n \le 25$

Re-linearization and elimination of variables in Boolean equation systems