An algorithm for computing Gröbner basis and
the complexity evaluation

Yong-Jin Kim\textsuperscript{[1,2]}, Hyon-Song Paek\textsuperscript{[2]}, Nam-Chol Kim\textsuperscript{[3,*]}, Chong-Il Byon\textsuperscript{[4]}

\textsuperscript{[1]} Natural Science Academy, Kim Il Sung University, Pyongyang, DPR of Korea
\textsuperscript{[2]} Faculty of Mathematics, Kim Il Sung University, Pyongyang, DPR of Korea
\textsuperscript{[3]} Faculty of Physics, Kim Il Sung University, Pyongyang, DPR of Korea
\textsuperscript{[4]} College of Computer Science, Kim Il Sung University, Pyongyang, DPR of Korea

* Corresponding author e-mail address: ryongnam19@yahoo.com

ABSTRACT
The first Gröbner basis algorithm was constructed by Buchberger in 1965; thus it bears his name to this day – Buchberger’s algorithm.\textsuperscript{[9]} Though Buchberger’s algorithm looks relatively simple, it can take a very large amount of time. The step that creates \( h_0 \) via a normal form calculation is computationally very difficult. This is particularly frustrating (and wasteful) if the normal form calculation results in \( h_0 = 0 \) because all that computation ends up adding nothing to the Gröbner basis (\( h_0 \) is only added if it is non-zero). It would be nice if there were some way to “see ahead” and eliminate critical pairs whose S-polynomials top-reduce to 0 rather than actually computing the normal form of those S-polynomials and getting 0. The person who find this method is Faugère.

In 2002, J.C. Faugère published an algorithm called F5 in [3]. This algorithm has been shown in empirical tests to be the fastest Gröbner-basis-generating algorithm devised. But this original version of F5 is given in programming codes, so it is a bit difficult to understand. So By Yao Sun and Dingkang Wang, the F5 algorithm is simplified as F5B in a Buchberger’s style such that it is easy to understand and implement in 2010.\textsuperscript{[10]} And in 2008 Justin Gash supposed F5t that is the advance of F5 in \textsuperscript{[6]}, in 1999 J.C. Faugere supposed F4 in [2], so many algorithms to construct Grobner bases suggested.

But with this, it is the important problem to construct an efficient algorithm which can be done F5-reduction more quickly, too.

In this paper, we suggest a new efficient algorithm in order to compute S-polynomial reduction rapidly in the known algorithm for computing Gröbner bases, and compare the complexity with others.

KEYWORDS
Gröbner basis; F5 algorithm; Buchberger algorithm

1. INTRODUCTION
First we introduce some algorithms for computing Gröbner basis. Let \( F \) be a finite subset of \( K[X] \), then we can compute Gröbner basis \( G \) in \( K[X] \) such that \( F \subseteq G, Id(G) = Id(F) \) as follows.

1.1 Buchberger algorithm\textsuperscript{[1,2,4,6,8,9]}
Given: \( F = \) a finite subset of \( K[X] \)
Find: \( G = \) a Gröbner basis in \( K[X] \) with \( F \subseteq G \) and \( Id(G) = Id(F) \)

\textbf{Begin:} \( G \leftarrow F \)
\begin{align*}
B & \leftarrow \{ \{g_i, g_j\} \mid g_i, g_j \in G, g_i \neq g_j \} \\
\text{While } B \neq \emptyset & \text{ do} \\
& \text{select } \{g_1, g_2\} \text{ from } B \\
& B \leftarrow B \setminus \{ \{g_1, g_2\} \}
\end{align*}


\[ h \leftarrow \text{spol}(g_1, g_2) \]
\[ h_0 \leftarrow \text{some normal form of } h \text{ modulo } G \]

If \( h_0 \neq 0 \) then
\[ B \leftarrow B \cup \{ \{g, h_0\} \mid g \in G \} \]
\[ G \leftarrow G \cup \{h_0\} \]

End if

End while

End

This algorithm takes a very large amount of time to reduce S-polynomial to 0.

To avoid this, many algorithms have been presented. F5B algorithm presented recently in [10] is one of the most efficient algorithms.

1.2 F5B algorithm\textsuperscript{[2,3,5,6,10,11]}

Input: \( f_i \in K[X], i = 1, m \)

Output: The Gröbner basis \( G \) of \( \text{ld}(f_1, \cdots, f_m) \)

Begin:
\[ F_i := (e_i, f_i), i = 1, m \]
\[ B := \{F_i \mid i = 1, m\} \]
\[ CP := \{[F_i, F_j] \mid 1 \leq i < j \leq m\} \]

While \( CP \neq \emptyset \) do

\( cp := \text{an element of } CP \)
\[ CP := CP \setminus \{cp\} \]

If (\( cp \) meets neither Syzygy Criterion nor Rewritten Criterion) then
\[ sp := \text{S-polynomial of } cp \]
\[ P := \text{the reduction result of } sp \text{ by } B \]

If \( \text{poly}(P) \neq 0 \) then
\[ CP := CP \cup \{[P, Q] \mid Q \in B\} \]

End if
\[ B := B \cup \{P\} \]

End if

End while

Return \( \{\text{poly}(Q) \mid Q \in B\} \)

End

The following is a subalgorithm of F5B algorithm to obtain the reduction result by \( B \)
1.3 Reduction algorithm\cite{3,5,6,10,11}

Input: a set of signed polynomials \textit{Todo} which need to be reduced, a set of signed polynomials \textit{B}

Output: a set of signed polynomials \textit{Done} in normal form by the set \textit{B}.

Begin:

\textit{Done} := \emptyset

\textbf{While} \textit{Todo} \neq \emptyset \textbf{do}

\begin{itemize}
  \item \textit{F} := the signed polynomial with minimal signature in set \textit{Todo}
  \item \textit{Todo} := \textit{Todo} \setminus \{\textit{F}\}
  \item (\textit{Done}',\textit{Todo}') := F5-reduction (\textit{F},\textit{B},\textit{Todo})
  \item \textit{Done} := \textit{Done} \cup \textit{Done}'
  \item \textit{Todo} := \textit{Todo} \cup \textit{Todo}'
\end{itemize}

\textbf{return} \textit{Done}

End while

End

The following is F5-reduction algorithm of above algorithm.

1.4 F5-reduction algorithm\cite{10}

Input: a signed polynomial \textit{F}, a set of signed polynomials \textit{B}

Output: a 2-tuple (\textit{Done},\textit{Todo})

Begin:

\begin{itemize}
  \item \textit{G} := a signed polynomial in \textit{B} such that
    \begin{enumerate}
      \item \textit{HT}(\textit{G}) \mid \textit{HT}(\textit{F}) \text{ and denote } \textit{v} := \frac{\textit{HM}(\textit{F})}{\textit{HM}(\textit{G})}
      \item signature \textit{sign}(\textit{F}) > \textit{sign}(\textit{v} \cdot \textit{G})
      \item \textit{v} \cdot \textit{G} \text{ is not divisible by } \textit{B}, \text{ and}
      \item \textit{v} \cdot \textit{G} \text{ is not rewritable by } \textit{B}
    \end{enumerate}
  \textbf{If} such \textit{G} does not exist \textbf{then}
    \textbf{return}(\{\textit{F}\},\emptyset)
  \textbf{Else}
    \textbf{return}(\emptyset,\{\textit{F} - \textit{vG}\})
\end{itemize}

End

So we put the research results of previous papers together, and then we will obtain the following conclusion conclusion: by this time, the research has been done which find the S-polynomials that can be reduced to 0 by using Syzygy Criterion and Rewrittern Criterion. Especially in F5B algorithm it is the important problem to construct an efficient algorithm which can be done F5-reduction more quickly.

In this paper we will establish the following problem such as:
First: we are going to suggest an efficient algorithm which can be done S-polynomial reduction quickly in the algorithms for computing Gröbner basis presented by this time.

Second: we are going to compare the complexity of the suggested algorithm and the previous algorithms for computing Gröbner basis.

2. AN ALGORITHM FOR COMPUTING GRÖBNER BASIS

In this section we focus our attention on making a pair-selection-method newly, and by using it constructing an algorithm for Gröbner basis updated the reduction sub-algorithm of F5B algorithm.

Let polynomial \( h \) need to be reduced by a set of polynomials \( G \). Here

\[
h = a_1X^{h_1} + a_2X^{h_2} + \cdots + a_pX^{h_p},
\]

\[
f = b_1X^{f_1} + b_2X^{f_2} + \cdots + b_qX^{f_q} \in G.
\]

When \( h \) is reduced by \( f \), if \( u = \frac{X^{h_1}}{X^{f_1}} \) then one reduction step of \( h \) by \( f \) is

\[
h - \frac{a_1}{b_1}uf = (a_1X^{h_1} + a_2X^{h_2} + \cdots + a_pX^{h_p}) - \frac{a_1}{b_1}(ub_1X^{f_1} + ub_2X^{f_2} + \cdots + ub_qX^{f_q}) =
\]

\[
a_2X^{h_2} + \cdots + a_pX^{h_p} - \frac{a_1}{b_1}b_2uX^{f_2} - \cdots - \frac{a_1}{b_1}b_quX^{f_q}
\]

In every reduction step, the deg of \( HT(h) \) becomes lower as fast as possible, \( h \) will be reduced through the most rapidly reduction step.

If \( h - uf = h' \) then \( HT(h') = \max \{ X^{h_2}, uX^{f_2} \} \), as the deg of \( uX^{f_2} \) becomes lower, \( h \) is reduced faster.

There are two cases with \( X^{h_2} \) and \( uX^{f_2} \).

Case1: \( X^{h_2} > uX^{f_2} \)

In this case \( HT(h') = X^{h_2} \), so in the next reduction step \( X^{h_2} \) is removed. And then let \( uX^{f_2} \) become a head term through the many reduction steps. From this, as the deg of \( uX^{f_2} \) is low, it is reduced faster.

Case2: \( X^{h_2} \leq uX^{f_2} \)

In this case \( HT(h') = uX^{f_2} \), as the deg of \( uX^{f_2} \) is low, it is reduced faster. So we can know that the deg of \( uX^{f_2} \) is as low as possible to be reduced rapidly. So let polynomial \( h = a_1X^{h_1} + a_2X^{h_2} + \cdots + a_pX^{h_p} \) need to be reduced by a set \( G \). If \( h \) is reduced by \( f_k \) satisfying the following conditions, then S-polynomial reduction has been done more quickly than F5B algorithm.

\[
f_i = b_{i_1}X^{h_1} + b_{i_2}X^{h_2} + \cdots + b_{i_q}X^{h_q}(i = 1, m) \in G
\]

(1) \( u_k = \frac{HT(h)}{HT(f_k)} \in T \)

(2) \( u_k \cdot X^{k_2} = \min \{ u_j \cdot X^{i_2} \} \)

Upon this we will construct the algorithm for computing Gröbner basis as following.
2.1 S-polynomial reduction algorithm
Input: a signed polynomial $sp \in K[X]$, a set of signed polynomials $B = \{F_1, \cdots, F_m\}$

Output: a signed polynomial $sp_0$ such that $sp \xrightarrow{B} sp_0$

Begin

$h := poly(sp) ;$

$f_i := poly(F_i) (i = 1, m) ;$

$G := \{f_1, \cdots, f_m\} ;$

$f_i := \text{an element of } G ;$

$t_{i,1} := HM(f_i) ;$

$t_{i,2} := \max(T(f_i) \setminus \{HT(f_i)\}) ;$

$k := 1 ;$

While $k \neq 0$ do

$k := \text{reduction sequence algorithm} (h, G) ;$

if $k = 0$ then

return $sp;$

else

$u := \frac{HT(h)}{HT(f_k)} ;$

$sp := sp - u \cdot F_k ;$

$h := poly(sp) ;$

End if

End while

Return $sp ;$

Following is the reduction sequence algorithm of above algorithm.

2.2 Reduction sequence algorithm $(h, G)$
Input: polynomial $h \in K[X], G = \{f_1, \cdots, f_m\}$

Output: index $k$ of $f_k$ which is chosen to reduce $h$ rapidly

Begin

$temp1 = 0, temp2 = HT(h) ;$

For $i = 1$ to $|G|$

If $HT(f_i) \mid HT(h)$ then

$u := \frac{HT(h)}{HT(f_i)}$
If \( u \cdot t_{i2} < temp2 \) then
\[
\text{temp2} = u \cdot t_{i2}
\]
\[
\text{temp1} = i
\]
End if
End if
End for
Return \( \text{temp1} \)

2.3 Correctness of algorithm
The algorithm can stop when it doesn’t satisfy the condition of the while-loop. In other words \( CP = \phi \).

The density of \( B \) isn’t over the number of all terms as possible at most. So after that the density of \( B \) doesn’t increase any more and on the contrary the density of \( CP \) decrease continuously. So algorithm stops exactly.

When algorithm stops, let a set of signed polynomials be \( B \). Then we know that S-polynomial reduction \( \text{spol}(F,G), B) = 0 \) for arbitrary signed polynomial \( F,G \in B \) from S-polynomial reduction algorithm. So the set of polynomials \( \{ \text{poly}(Q) | Q \in B \} \) becomes a Gröbner basis of \( 1d(f_1, \cdots, f_m) \subset K[X] \).

3. COMPLEXITY COMPARITION AND EVALUATION
In this section we compare the complexities of Buchberger algorithm, F5B algorithm and the algorithm that we suggest. The complexity evaluation is done following the algorithm.[7]

**Input of algorithm:** \( F = \{ f_1, \cdots, f_m \} \subseteq K[X_1, \cdots, X_n] \)

**Output of algorithm:** Gröbner basis \( G \) of \( 1d(f_1, \cdots, f_m) \)

**Complexity evaluation measure:** an operation in the ground field is counted as one step.

**Goal:** Calculation the number of steps [9]
\[
D := \max \deg(h_i | h_i \in H), H \text{ : a set of occuring polynomials during the computation}
\]
\[
D \leq (8 \cdot \max \deg(F) + 1) \cdot 2^{\min \deg(F)} [9]
\]

Here \( \max \deg(F) \) means maximum in the degs of polynomials of \( F \), \( \min \deg(F) \) means minimum.

The number of terms of \( n \)-variables, D-degree polynomial \( f \) is
\[
N(D,n) = \sum_{k=0}^{D} \binom{n + k - 1}{k} = \sum_{k=0}^{D} \binom{n + k - 1}{n - 1} = \binom{n + D - 1 + 1}{n - 1 + 1} = \binom{n + D}{n}.
\]

3.1 Complexity of Buchberger algorithm

**Proposition 1.** The complexity of Buchberger algorithm is
\[
\frac{3}{2} n \cdot N(D,n)^3 + N(D,n)^4 + (2mn - \frac{1}{2} n + 7) \cdot N(D,n)^3 + \\
+ (-\frac{3}{2} n - \frac{3}{2}) \cdot N(D,n)^2 + (-\frac{1}{2} m^2 n - mn - \frac{3}{2} n - \frac{1}{2}) \cdot N(D,n)
\]

**Proof:**

Begin:
\[
G := \{ f_1, \cdots, f_m \}
\]
\[
B := \{ (g_1, g_2), \cdots, (g_{m-1}, g_m) \}
\]

While \( B \neq \phi \) do

Let's begin our discussion by focusing our attention to \((i+1)\)th loop.

- select \((g_a, g_b)\) from \(B_i\)
- \(B_i := B_i \setminus \{g_a, g_b\}\)
- The complexity of \(h = spol(g_a, g_b)\)
  - The complexity of finding the head term \(t_a = HT(g_a)\) in \(N(D,n)\) terms of \(g_a\) is \(2n \cdot N(D,n)\).
  - The complexity of finding the head term \(t_b = HT(g_b)\) in \(N(D,n)\) terms of \(g_b\) is \(2n \cdot N(D,n)\).
  - The complexity of computing \(lcm(t_a, t_b) = t\) is \(n\).
  - The complexity of computing \(s_a\) such that \(t = s_a t_a\) is \(n\).
  - The complexity of computing \(s_b\) such that \(t = s_b t_b\) is \(n\).
  - The complexity of computing \(spol(g_a, g_b) = c_b s_a g_a - c_a s_b g_b\) is \((2n+3) \cdot N(D,n) + n \cdot N(D,n)^2\).
    - The complexity of computing \(c_b s_a g_a\) is \((n+1) \cdot N(D,n)\).
    - The complexity of computing \(c_a s_b g_b\) is \((n+1) \cdot N(D,n)\).
    - The complexity of computing \(c_b s_a g_a - c_a s_b g_b\) is \(n \cdot N(D,n)^2 + N(D,n)\).

So the complexity of computing \(h = spol(g_a, g_b)\) is
\[
n \cdot N(D,n)^2 + (6n+3) \cdot N(D,n) + 3n
\]

- The complexity of computing \(h \xrightarrow{g} h_0\)
  - The complexity of computing \(HT(g)\) for any \(g\) in \(G_i\) is \(2n \cdot N(D,n) \cdot |G_i|\)
  - The complexity of computing \(HT(h)\) is \(2n \cdot N(D,n)\)
  - The complexity of computing \(g\) such as \(HT(g)\) in \(G_i\) is \(n \cdot |G_i|\).
  - The complexity of computing \(v_{k_i}\) such as \(HT(h) = v_{k_i} \cdot HT(g_{k_i})\) is \(n\).
The complexity of computing \( h \mapsto h - \frac{HC(h)}{HC(g_{k_i})} v_{k_i} g_{k_i} = h \) is

\[ n \cdot N(D, n)^2 + (n + 2) \cdot N(D, n) + 1 \]

\( \checkmark \) The complexity of computing \( \frac{HC(h)}{HC(g_{k_i})} v_{k_i} g_{k_i} \) is \( 1 + (1 + n) \cdot N(D, n) \).

\( \checkmark \) The complexity of computing \( h - \frac{HC(h)}{HC(g_{k_i})} v_{k_i} g_{k_i} \) is \( n \cdot N(D, n)^2 + N(D, n) \).

\[ \ldots \ldots \]

So the complexity of computing \( h \mapsto h \) is

\[ N(D, n) \cdot (n \cdot N(D, n)^2 + (2n \cdot N(D, n) + n) \cdot |G_i| + (3n + 2) \cdot N(D, n) + n + 1) = n \cdot N(D, n)^3 + (2n \cdot N(D, n)^2 + n \cdot N(D, n)) \cdot |G_i| + (3n + 2) \cdot N(D, n)^2 + (n + 1) \cdot N(D, n) \] (2)

- The complexity of computing “if \( h_0 \neq 0 \) then” is 1. (3)

\( B_{i+1} = B_i \cup \{ g, h_0 \} \mid g \in G_i \}

\( G_{i+1} = G_i \cup \{ h_0 \} \}

End if

So the complexity of computing \((i + 1)\)th loop is

\[ T_i = (1) + (2) + (3) = n \cdot N(D, n)^3 + (2n \cdot N(D, n)^2 + n \cdot N(D, n)) \cdot |G_i| + (4n + 2) \cdot N(D, n)^2 + (7n + 4) \cdot N(D, n) + 3n + 1 \]

End while

So all complexity is \( T = \sum_{i=1}^{w} T_i \).

Here \( W \) is the number of while-loop of algorithm.

End.

Now let’s compute \( |G_i| \) and \( W \).

\( |G_0| = m, |B_0| = \frac{m(m-1)}{2} \) and the number of while-loop is \( W = (N(D, n) - m) + |B_{N(D, n)-m}| \).

For \( i \), such as \( 1 \leq i \leq (N(D, n) - m) + |B_{N(D, n)-m}| \), we will go through in two cases as following.

**Case 1:** \( 1 \leq i \leq N(D, n) - m \)

\[ |G_i| = m + i \]

\[ |B_{i+1}| = |B_i| - 1 + |G_i| = |B_i| + (m + i - 1) \]
Let $A(x) = \sum_{i \geq 0} B_i \cdot x^i$.

Let’s multiply both sides by $x^i$ and sum up both sides.

$$\sum_{i \geq 0} B_{i+1} \cdot x^i = \sum_{i \geq 0} B_i \cdot x^i + \sum_{i \geq 0} i \cdot x^i + (m-1) \sum_{i \geq 0} x^i.$$  

$$\frac{1}{x} (A(x) - A(0)) = A(x) + \frac{x}{(1-x)^2} + (m-1) \frac{1}{x-1} \left( \frac{1}{x} - 1 \right) A(x) = \frac{1}{x} \frac{m(m-1)}{2} + \frac{x}{(1-x)^2} + (m-1) \frac{1}{1-x}$$

$$A(x) = \frac{m(m-1)}{2} \frac{1}{1-x} + \frac{x^2}{(1-x)^3} + (m-1) \frac{x}{1-x} =$$

$$= \frac{m(m-1)}{2} \sum_{i \geq 0} x^i + \sum_{i \geq 0} \frac{i(i-1)}{2} x^i + \sum_{i \geq 0} (m-1)i x^i =$$

$$= \sum_{i \geq 0} \left( \frac{m(m-1)}{2} + (m-1)i + \frac{i(i-1)}{2} \right) x^i$$

$$\therefore B_i = \frac{m(m-1)}{2} + (m-1)i + \frac{i(i-1)}{2}$$

$$\therefore B_{N(D,n)-m} = \frac{(N(D,n) - m - 1)(N(D,n) - m)}{2} + (m-1)(N(D,n) - m) + \frac{m(m-1)}{2}$$

$$= \frac{1}{2} N(D,n)^2 - \frac{3}{2} N(D,n) + m$$

Case 2: $N(D,n) - m + 1 \leq i \leq W$

$$| G_i | = N(D,n)$$

$$| B_i | \geq | B_{i-1} | - 1$$

$$\therefore T = \sum_{i=1}^{W} T_i = \sum_{i=1}^{N(D,n)-m} T_i + \sum_{i=N(D,n)-m+1}^{W} T_i =$$

$$= \sum_{i=1}^{N(D,n)-m} \left[ n \cdot N(D,n)^3 + (2n \cdot N(D,n)^2 + n \cdot N(D,n)) \cdot | G_i | + (4n+2) \cdot N(D,n)^2 + (7n+4) \cdot N(D,n) + 3n+1 \right]$$

$$+ \left[ B_{N(D,n)-m} \left[ n \cdot N(D,n)^3 + (2n \cdot N(D,n)^2 + n \cdot N(D,n)) \cdot | G_i | + (4n+2) \cdot N(D,n)^2 + (7n+4) \cdot N(D,n) + 3n+1 \right] =$$

$$= \frac{3}{2} n \cdot N(D,n)^5 + N(D,n)^4 + (2mn - \frac{1}{2} \cdot n + 7) \cdot N(D,n)^3$$

$$+ (\frac{3}{2} n - 3) \cdot N(D,n)^2 + (\frac{1}{2} m^2 - mn - \frac{3}{2} n - \frac{1}{2}) \cdot N(D,n)$$

So the complexity of Buchberger algorithm is

$$T = \frac{3}{2} n \cdot N(D,n)^5 + N(D,n)^4 + (2mn - \frac{1}{2} \cdot n + 7) \cdot N(D,n)^3 +$$

$$+ (\frac{3}{2} n - 3) \cdot N(D,n)^2 + (\frac{1}{2} m^2 - mn - \frac{3}{2} n - \frac{1}{2}) \cdot N(D,n)$$

This completes the proof. □
3.2 Complexity of F5B algorithm

Proposition 2. The complexity of F5B algorithm is

\[
\begin{align*}
&\frac{2n}{3}N(D, n)^3 + (mn + \frac{14}{3}n + \frac{2}{3})N(D, n)^2 + (-mn + m + \frac{11}{6})N(D, n)^2 + \\
&\quad + (\frac{10}{3}m^3n - 3m^2n - m^2 + \frac{14}{3}mn + 2m - \frac{16}{3}n - \frac{2}{3})N(D, n)^2 + \\
&\quad + (\frac{2}{3}mn - \frac{2}{3}m^3 + m^2n + \frac{17}{6}mn + \frac{8}{3} - \frac{11}{2}n - 2)N(D, n) + \\
&\quad + (\frac{11}{2}m^2n - 4m^2 + \frac{7}{2}mn + 2m)
\end{align*}
\]

Proof:

Begin:

- \( i = 1, m, F_i = (e_i, f_i) \)
- \( B_0 = \{F_1, \ldots, F_m\} \)
- \( CP_0 := \{[F_i, F_j] | 1 \leq i < j \leq m\} \)
  (\( CP_0 = \{[F_1, F_2], \ldots, [F_{m-1}, F_m]\} \))
  - The complexity of computing \([F_i, F_j] = [u, F_i, v, F_j]\) is \(4n \cdot N(D, n) + 3n\).
    ✓ The complexity of computing is \(t_i = HT(poly(F_i))\) is \(2n \cdot N(D, n)\).
    ✓ The complexity of computing is \(t_j = HT(poly(F_j))\) is \(2n \cdot N(D, n)\).
    ✓ The complexity of computing \(lcm(t_i, t_j) = t\) is \(n\).
    ✓ The complexity of computing \(u\) such as \(t = ut_i\) is \(n\).
    ✓ The complexity of computing \(v\) such as \(t = vt_j\) is \(n\).

So the complexity of computing \(CP_0\) is

\[
\frac{m(m-1)}{2}(4n \cdot N(D, n) + 3n)
\]

- While \(CP_i \neq \emptyset\) do

  Let’s begin our discussion by focusing our attention to \(i\) – th loop.

  - \(cp \leftarrow CP_{i-1}\)
  - \(CP_i \leftarrow CP_{i-1} \setminus \{cp\}\)
  - If \(cp\) meets neither Syzygy Criterion nor Rewritten Criterion, then
    - \(cp\) meets Syzygy Criterion?
      - \(cp = (u, F, v, G)\)
        ✓ \(uF\) is divisible?
          The complexity of computing \(u \cdot S(F)\) is \(n\)
          The complexity of computing \(HT(g)\) is \(2n \cdot N(D, n)\).
The complexity of computing \( HT(g) | u \cdot S(F) \) is \( n \).

The complexity of computing \( i < j \) is 1

So for any \( g \) in \( B_{i-1} \), the complexity is \( (2n \cdot N(D,n)+n+1) \cdot |B_{i-1}| \).

The complexity of computing “\( uF \) is divisible?” is \( (2n \cdot N(D,n)+n+1) \cdot |B_{i-1}| + n \).

✓ The complexity of computing “\( vG: \)is divisible?” is \( (2n \cdot N(D,n)+n+1) \cdot |B_{i-1}| + n \).

• So the complexity of computing “\( cp: \) meets Syzygy Criterion?” is \( 2((2n \cdot N(D,n)+n+1) \cdot |B_{i-1}| + n) \).

• \( cp \) meets Rewritten Criterion?
  ✓ \( uF \) is rewritable?

The complexity of computing \( u \cdot S(F) \) is \( n \).

The complexity of computing \( S(G) | u \cdot S(F) \) for \( G \) in \( G \) is \( n \).

The complexity of computing “\( G \) is generated later than \( F \)” is 1.

So for any \( g \) in \( B_{i-1} \), the complexity is \( (n+1) \cdot |B_{i-1}| \).

So the complexity of computing “\( uF \) is rewritable?” is \( n + (n+1) \cdot |B_{i-1}| \).

✓ The complexity of computing “\( vG \) is rewritable?” is \( n + (n+1) \cdot |B_{i-1}| \).

The complexity of computing “\( cp \) meets Rewritten Criterion?” is \( 2(n + (n+1)) \cdot |B_{i-1}| \).

So the complexity of computing “if-condition satisfy?” is

\[
2((2n \cdot N(D,n)+n+1) \cdot |B_{i-1}| + n) + 2(n + (n+1) \cdot |B_{i-1}|) = 2((2n \cdot N(D,n)+2n+2) \cdot |B_{i-1}| + 2n) = (4n \cdot N(D,n) + 4n + 4) \cdot |B_{i-1}| + 4n \tag{5}
\]

then

• \( sp \leftarrow \) S-polynomial of \( cp \)

\( cp = (u, F, v, G) = (u, (e_i, f), v, (e_j, g)) \)

\( spol(cp) = HC(g)uF - HC(f)vG = (HC(g)ue_i, HC(g)uf) - (HC(f)ve_j, HC(f)vg) \)

✓ \( HC(g)uF = (HC(g)ue_i, HC(g)uf) : (n+1)m + (n+1) \cdot N(D,n) \)

✓ \( HC(f)vG = (HC(f)ve_j, HC(f)vg) : (n+1)m + (n+1) \cdot N(D,n) \)

\( sp = HC(g)uF - HC(f)vG = \)

\( = (HC(g)ue_i, HC(g)uf) - (HC(f)ve_j, HC(f)vg) : (n \cdot N(D,n)^2 + N(D,n)) \cdot m + + n \cdot N(D,n)^2 + N(D,n) \)

The complexity of computing \( sp \) is

\[
2((n + 1)m + (n+1) \cdot N(D,n)) + (n \cdot N(D,n)^2 + N(D,n)) \cdot m + + n \cdot N(D,n)^2 + N(D,n) = (mn + n) \cdot N(D,n)^2 + (2n + m + 3) \cdot N(D,n) + + 2(n+1)m \tag{6}
\]

• \( p \leftarrow \) reduction result of \( sp \) (\( sp \xrightarrow{B_{i-1}} p \))
\( \text{Done} \leftarrow \phi; \)

\( \text{While} \ Todo \neq \phi \ \text{do} \)

The complexity of computing “\( F \leftarrow \text{the signed polynomial with minimal signature in set } Todo \)” is \(|B_{i-1}| \cdot n\).

\( Todo \leftarrow Todo \setminus \{F\} \)

\((\text{Done}', Todo') \leftarrow F5\text{–reduction } (F, B_{i-1})\)

\( G \leftarrow B_{i-1} \)

(1) The complexity of computing “\( HT(G) \mid HT(F), v = \frac{HM(f)}{HM(G)} \)” is \(4n \cdot N(D, n) + 2n\).

The complexity of computing “\( HT(F) \)” is \(2n \cdot N(D, n)\).

The complexity of computing “\( HT(G) \)” is \(2n \cdot N(D, n)\).

The complexity of computing “\( HT(G) \mid HT(F) \)” is \(n\).

The complexity of computing “\( v = \frac{HM(f)}{HM(G)} \)” is \(n\).

(2) The complexity of computing “\( S(F) > S(v \cdot G) \)” is \(2n\)

The complexity of computing “\( S(v \cdot G) \)” is \(n\)

The complexity of computing “\( S(F) > S(v \cdot G) \)” is \(n\)

(3) The complexity of computing “\( v \cdot G \) is not divisible by \( B_{i-1} \)” is \((2n \cdot N(D, n) + n + 1) \cdot |B_{i-1}| + n\).

(4) The complexity of computing “\( v \cdot G \) is not rewritable by \( B_{i-1} \)” is \(n + (n + 1) \cdot |B_{i-1}|\).

\( \therefore \ (4n \cdot N(D, n) + 2n + 2n + (2n \cdot N(D, n) + n + 1) \cdot |B_{i-1}| + n + n + (n + 1) \cdot |B_{i-1}|) \cdot |B_{i-1}| = \)

\( = (2n \cdot N(D, n) + 2n + 2) \cdot |B_{i-1}|^2 + 4n \cdot N(D, n) \cdot |B_{i-1}| + 6n \cdot |B_{i-1}| \)

If such \( G \) does not exist

\( \text{then} \ \text{return} \ \{(F), \phi\} \)

\( \text{else} \ \text{return} \ \{(\phi, (F \cdot v \cdot G)), \}

\( mn \cdot N(D, n) + n \cdot N(D, n)^2 + N(D, n)) \cdot m + n \cdot N(D, n)^2 + N(D, n) \)

The complexity of computing \( v \cdot G \) is \(mn \cdot N(D, n)\).

The complexity of computing \( F \text{ – } v \cdot G \) is

\( (n \cdot N(D, n)^2 + N(D, n)) \cdot m + n \cdot N(D, n)^2 + N(D, n) \).

End if

\( \text{Done} \leftarrow \text{Done} \cup \text{Done}' \)

\( Todo \leftarrow Todo \cup Todo' \)

\( \checkmark \ \text{End while} \)

\( \checkmark \ \text{return} \ \text{Done} \)

The complexity of F5-reduction algorithm is
\[ \{ |B_{i-1}| n + (2n \cdot N(D, n) + 2n + 2) \cdot |B_{i-1}|^2 + 4n \cdot N(D, n) \cdot |B_{i-1}| + 6n \cdot |B_{i-1}| + mn \cdot N(D, n) + \] \\
+ (n \cdot N(D, n)^2 + N(D, n)) \cdot m + n \cdot N(D, n)^2 + N(D, n) \} \cdot N(D, n) = \\
= (2n \cdot N(D, n)^2 + (2n + 2) \cdot N(D, n)) \cdot |B_{i-1}| + (4n \cdot N(D, n)^2 + 7n \cdot N(D, n)) \cdot |B_{i-1}| + \\
+ (mn + n) \cdot N(D, n)^3 + (mn + m + 1)N(D, n)^2 \] 

\[ \text{if } \text{poly}(P) \neq 0 \text{ then} \]

The complexity of computing “\( CP_i \leftarrow CP_i \cup \{ [P, Q] \mid Q \in B_{i-1} \} \)” is

\[ (4n \cdot N(D, n) + 3n) \cdot |B_{i-1}|. \] 

\[ \text{End if} \]

\[ \bullet B_i \leftarrow B_{i-1} \cup \{ P \} \]

\[ \text{End if} \]

The complexity of \( i \)-th loop is

\[ T_i = (5) + (6) + (7) + (8) \]

\[ = (4n \cdot N(D, n) + 4n + 4) \cdot |B_{i-1}| + 4n + (mn + n) \cdot N(D, n)^2 + \]

\[ + (2n + m + 3) \cdot N(D, n) + 2(n + 1)m + (2n \cdot N(D, n)^2 + \]

\[ + (2n + 2) \cdot N(D, n)) \cdot |B_{i-1}|^2 + (4n \cdot N(D, n)^2 + 7n \cdot N(D, n)) \cdot |B_{i-1}| + (mn + n) \cdot N(D, n)^3 + \]

\[ + (mn + m + 1) \cdot N(D, n)^2 + (4n \cdot N(D, n) + 3n) \cdot |B_{i-1}| = \]

\[ = (2n \cdot N(D, n)^2 + (2n + 2) \cdot N(D, n)) \cdot |B_{i-1}|^2 + (4n \cdot N(D, n)^2 + 15n \cdot N(D, n) + 7n + 4) \cdot |B_{i-1}| + \]

\[ + (mn + n) \cdot N(D, n)^3 + (2mn + m + n + 1) \cdot N(D, n)^2 + (2n + m + 3) \cdot N(D, n) + 4n + 2(n + 1)m \]

\[ \text{End while} \]

\[ \text{- return } \{ \text{poly}(Q) \mid Q \in B_i \} \]

\[ T = (4) + \sum_{i=1}^{w} T_i = \frac{m(m-1)}{2} \cdot (4n \cdot N(D, n) + 3n) + \sum_{i=1}^{w} T_i \]

End

Now let’s compute \( |G_i| \) and \( w \).

\[ \begin{cases} 
|B_0| = m \\
|CP_0| = \frac{m(m-1)}{2} \\
|B_i| = |B_{i-1}| + 1 = m + i \\
|CP_i| = |CP_{i-1}| - 1 + |B_{i-1}| - |CP_{i-1}| - 1 + m + i - 1 \\
|CP_i| = \frac{i(i-1)}{2} + (m-1)i + \frac{m(m-1)}{2}, 1 \leq i \leq W 
\end{cases} \]

The number of while-loop is \( W = (N(D, n) - m) + |CP_{N(D,n)-m}| \).

For \( i \) such as \( 1 \leq i \leq (N(D, n) - m) + |B_{N(D,n)-m}| \) we will go through in two cases as following.

Case 1: \( 1 \leq i \leq N(D, n) - m \), \( |B_{i-1}| = m + i - 1 \)
\[ T_i = (2n \cdot N(D,n)^2 + (2n + 2) \cdot N(D,n)) \cdot |B_{i-1}|^2 + (4n \cdot N(D,n)^2 + 15n \cdot N(D,n) + 7n + 4) \cdot |B_{i-1}| +
+(mn + n) \cdot N(D,n)^3 + (2mn + m + n + 1) \cdot N(D,n)^2 + (2n + m + 3) \cdot N(D,n) + 4n + 2(n + 1)m =
= (2n \cdot N(D,n)^2 + (2n + 2) \cdot N(D,n)) \cdot i^2 +
+(4mn \cdot N(D,n)^2 + (4mn + 4m + 11n - 4) \cdot N(D,n) + 7n + 4)i
+(mn + n) \cdot N(D,n)^3 + (2m^2n + 2mn + m - n + 1) \cdot N(D,n)^2
+(2m^2n + 11mn + 2m^2 - 11n - 3m + 5) \cdot N(D,n) +
+9mn + 6m - 3n - 4 \cdot (N(D,n) - m) =
\]
\[
N(D,n)^m \sum_{i=1}^{N(D,n)-m} T_i
\]
\[ = (2n \cdot N(D,n)^2 + (2n + 2) \cdot N(D,n)) \cdot \sum_{i=1}^{N(D,n)-m} i^2 + (4mn \cdot N(D,n)^2 +
+(4mn + 4m + 11n - 4) \cdot N(D,n) + 7n + 4)i
+(mn + n) \cdot N(D,n)^3 + (2m^2n + 2mn + m - n + 1) \cdot N(D,n)^2
+(2m^2n + 11mn + 2m^2 - 11n - 3m + 5) \cdot N(D,n) +
+9mn + 6m - 3n - 4 \cdot (N(D,n) - m) =
\]
\[
= \frac{2n}{3} \cdot N(D,n)^5 + (mn + \frac{8}{3}n + \frac{2}{3}n^2) \cdot N(D,n)^4 + (-m^2n + mn + m + \frac{35}{6}n)N(D,n)^3 +
+(\frac{10}{3}m^3n - 3m^2n - m^2 + \frac{2}{3}mn - \frac{4}{3}n + \frac{16}{3})N(D,n)^2 +
+(-\frac{2}{3}m^3n - \frac{2}{3}m^2n - m^2n + \frac{5}{6}mn - \frac{4}{3}n + \frac{1}{2}n - 2)N(D,n) +
+(-\frac{11}{2}m^2n - 4m^2 - \frac{1}{2}mn + 2m)
\]
(9)

Case2: \(N(D,n) - m + 1 \leq i \leq W\)

\[ |B_{i-1}| = N(D,n) \]
\[ T_i = (4n \cdot N(D,n) + 4n + 4) \cdot |B_{i-1}| + 4n = 4n \cdot N(D,n)^2 + (4n + 4) \cdot N(D,n) + 4n \]

\[ \therefore \sum_{i=N(D,n)-m}^{W} T_i = \left[ \frac{(N(D,n) - m)(N(D,n) - m + 1)}{2} + (m - 1)(N(D,n) - m) + \frac{m(m - 1)}{2} \right] \cdot
\]
\[ (4n \cdot N(D,n)^2 + (4n + 4) \cdot N(D,n) + 4n) \]
\[ = 2n \cdot N(D,n)^4 - (4n - 2) \cdot N(D,n)^3 + (4mn - 4n - 6) \cdot N(D,n)^2 +
+ (4mn + 4m - 6n) \cdot N(D,n) + 4mn \]
(10)
\[ T = \frac{m(m-1)}{2} \cdot (4n \cdot N(0, n) + 3n) + (9) + (10) \]

\[ = \frac{2n}{3} \cdot N(D, n)^5 + (mn + \frac{14}{3} n + \frac{2}{3}) \cdot N(D, n)^4 + (-m^2 n + mn + m + \frac{11}{6} n + 2) \cdot N(D, n)^3 \]

\[ + (\frac{10}{3} m^3 n - 3m^2 n - m^2 + \frac{14}{3} mn + 2m - \frac{16}{3} n - \frac{2}{3}) \cdot N(D, n)^2 \]

\[ + (-\frac{2}{3} m^3 n - \frac{2}{3} m^3 + m^2 n + \frac{17}{6} mn + \frac{8}{3} m - \frac{11}{2} n - 2) \cdot N(D, n) \]

\[ + (-\frac{11}{2} m^2 n - 4m^2 + \frac{7}{2} mn + 2m) \]

So the complexity of F5B algorithm is

\[ T = \frac{2n}{3} \cdot N(D, n)^5 + (mn + \frac{14}{3} n + \frac{2}{3}) \cdot N(D, n)^4 + (-m^2 n + mn + m + \frac{11}{6} n + 2) \cdot N(D, n)^3 \]

\[ + (\frac{10}{3} m^3 n - 3m^2 n - m^2 + \frac{14}{3} mn + 2m - \frac{16}{3} n - \frac{2}{3}) \cdot N(D, n)^2 \]

\[ + (-\frac{2}{3} m^3 n - \frac{2}{3} m^3 + m^2 n + \frac{17}{6} mn + \frac{8}{3} m - \frac{11}{2} n - 2) \cdot N(D, n) + \]

\[ + (-\frac{11}{2} m^2 n - 4m^2 + \frac{7}{2} mn + 2m) \]

This completes the proof. \[\Box\]

3.3 Complexity of new algorithm

**Proposition 3.** The complexity of S-polynomial reduction algorithm is

\[ (mn + 4n) \cdot N(D, n)^4 + (-m^2 n + mn + m + \frac{15}{2} n + 31) \cdot N(D, n)^3 \]

\[ + (-3m^2 n - m^2 + 5mn - 5n - 1) \cdot N(D, n)^2 \]

\[ + (-7m^2 n + 10mn - m^2 - m - 2n - 2) \cdot N(D, n) + 4m^2 n - 2m^2 + 2m \]

**Proof:**

\[\triangleleft\]

**S-polynomial reduction algorithm**

Input : a signed polynomial \( sp \in K[X] \), a set of signed polynomials \( B = \{ F_1, \ldots, F_m \} \)

Output : a signed polynomial \( sp_0 \) such that \( sp \xrightarrow{\ast}_B sp_0 \)

**Begin**

- \( h \leftarrow \text{poly}(sp) \);

- \( f_i \leftarrow \text{poly}(F_i)(i = 1 \text{ to } m) \);

- \( G \leftarrow (f_1, \ldots, f_m) \)
  - \( f_i \) : an element of \( G \)
  - The complexity of computing \( t_{1i} := \text{HM}(f_i) \) is \( 2n \cdot N(D, n) \)
The complexity of computing is \( t_{i2} := \max(T(f_i) \setminus \{HT(f_i)\}) \) is \( 2n \cdot N(D,n) \).

- The complexity of computing \( \text{While } k \neq 0 \text{ do } \) is 1.
  - The complexity of computing \( \text{if } k \neq 0 \text{ then } \) is 1.
  - The complexity of computing \( \text{return } \) is \( n \).

The complexity of computing \( u := \frac{HM(h)}{HM(f_k)} \) is \( n \).

The complexity of computing \( sp := sp - u \cdot F_k \) is

\[
mn \cdot N(D,n) + (n \cdot N(D,n)^2 + N(D,n) \cdot m + n \cdot n \cdot N(D,n)^2 + N(D,n).
\]

\( h := \text{poly}(sp) \)

Endwhile

So the complexity of while-loop is

\[
[(2n \cdot N(D,n) + 3n) \cdot |B_{i-1}| + (mn + n) \cdot N(D,n)^2 + (mn + m + 2n + 1) \cdot N(D,n) + n + 2] \cdot N(D,n).
\]

- Return \( sp \)

So the complexity of S-polynomial reduction algorithm is

\[
(2n \cdot N(D,n)^2 + 7n \cdot N(D,n)) \cdot |B_{i-1}| + (mn + n) \cdot N(D,n)^3 +
\]

\[
+ (mn + m + 2n + 1) \cdot N(D,n)^2 + (n + 2) \cdot N(D,n) \tag{11}
\]

\[\Delta\] Reduction sequence algorithm \( (h,G) \)

\textbf{Input}: polynomial \( h \in K[X], G = \{f_1, \cdots, f_m\} \)

\textbf{Output}: index \( k \) of \( f_k \) which is chosen to reduce \( h \) rapidly

\textbf{Begin}

- \textbf{temp1} = 0, The complexity of computing \( \text{temp2} = HT(h) \) is \( 2n \cdot N(D,n) \).

- For \( i = 1 \) to \( |G| \)
  - The complexity of computing \( \text{if } HT(f_i) \mid HT(h) \text{ then } \) is \( 2n \cdot N(D,n) + n \).
    
    \[
u := \frac{HT(h)}{HT(f_i)}
\]

    The complexity of computing

    \( \text{if } u \cdot t_{i2} \leq \text{temp2} \text{ then } \) is \( 2n \).

    \[
temp2 = u \cdot t_{i2}
\]

    \[
temp1 = i
\]

End if
End if
End for

The complexity of for-while is \((2n \cdot N(D,n) + 3n) \cdot |B_{i-1}|\).

- Return temp
End

So the complexity of reduction sequence algorithm is \((2n \cdot N(D,n) + 3n) \cdot |B_{i-1}| + 2n \cdot N(D,n)\).

When S-polynomial reduction algorithm is used instead of F5-reduction algorithm of F5B algorithm, it’s complexity is as following.

The complexity of i-th loop is

\[
T_i = (5) + (6) + (11) + (8) \\
= (4n \cdot N(D,n) + 4n + 4) \cdot |B_{i-1}| + 4n + + (mn + n) \cdot N(D,n)^2 + (2n + m + 3) \cdot N(D,n) + \\
+ 2(n + 1)m + (2n \cdot N(D,n)^2 + 7n \cdot N(D,n)) \cdot |B_{i-1}| + (mn + n) \cdot N(D,n)^3 + (mn + m + 2n + 1) \cdot N(D,n)^2 + \\
+ (n + 2) \cdot N(D,n) + (4n \cdot N(D,n) + 3n) \cdot |B_{i-1}| = \\
= (2n \cdot N(D,n)^2 + 15n \cdot N(D,n) + 7n + 4) \cdot |B_{i-1}| + (mn + n) \cdot N(D,n)^3 + (2mn + m + 3n + 1) \cdot N(D,n)^2 \\
+ (3n + m + 5) \cdot N(D,n) + 2mn + 6n
\]

\[
T = (4) + \sum_{i=1}^{w} T_i = \frac{m(m-1)}{2} \cdot (4n \cdot N(D,n) + 3n) + \sum_{i=1}^{w} T_i
\]

\[
\left\{ \begin{array}{l}
|B_0| = \frac{m(m-1)}{2} \\
|CP_0| = \frac{m(m-1)}{2} \\
|B_i| = \frac{m+i}{2} \\
|B_{i-1}| = \frac{m+i-1}{2} \\
|CP_i| = |CP_{i-1}| + 1 \quad |B_{i-1}| = |CP_{i-1}| + 1 \quad |B_{i-1}| = |CP_{i-1}| + 1
\end{array} \right.
\]

\[W = (N(D,n) - m) + |CP_{N(D,n)}-m|\]

(1) \(1 \leq i \leq N(D,n) - m\)

\[|B_{i-1}| = m + i - 1\]

\[
T_i = (2n \cdot N(D,n)^2 + 15n \cdot N(D,n) + 7n + 4) \cdot |B_{i-1}| + (mn + n) \cdot N(D,n)^3 + \\
+ (2mn + m + 3n + 1) \cdot N(D,n)^2 + (3n + m + 5) \cdot N(D,n) + 2mn + 6n = \\
= (2n \cdot N(D,n)^2 + 15n \cdot N(D,n) + 7n + 4) \cdot (m + i - 1) + (mn + n) \cdot N(D,n)^3 + \\
+ (2mn + m + 3n + 1) \cdot N(D,n)^2 + (3n + m + 5) \cdot N(D,n) + 2mn + 6n = \\
= (2n \cdot N(D,n)^2 + 15n \cdot N(D,n) + 7n + 4) \cdot i + (2n \cdot N(D,n)^2 + 15n \cdot N(D,n) + 7n + 4) \cdot (m - 1) \\
+ (mn + n) \cdot N(D,n)^3 + (2mn + m + 3n + 1) \cdot N(D,n)^2 + (3n + m + 5) \cdot N(D,n) + 2mn + 6n = \\
= (2n \cdot N(D,n)^2 + 15n \cdot N(D,n) + 7n + 4) \cdot i + (mn + n) \cdot N(D,n)^3 + (4mn + m + n + 1) \cdot N(D,n)^2 + \\
+ (15mn - 12n + m + 3) \cdot N(D,n) + 9mn + 4m - n - 4
\]
\[
\sum_{i=1}^{N(D,n)-m} T_i = \\
(2n \cdot N(D,n)^2 + 15n \cdot N(D,n) + 7n + 4) \cdot \sum_{i=1}^{N(D,n)-m} i + \\
(N(D,n) - m) \cdot ((mn + n) \cdot N(D,n)^3 + (4mn + m + n + 1) \cdot N(D,n)^2 \\
+ (15mn - 12n + m + 3) \cdot N(D,n) + 9mn + 4m - n - 4) = \\
(2n \cdot N(D,n)^2 + 15n \cdot N(D,n) + 7n + 4) \cdot \frac{1}{2} (N(D,n) - m)(N(D,n) - m + 1) \\
+ ((mn + n) \cdot N(D,n)^4 + (-m^2n + 3mn + m + n + 1) \cdot N(D,n)^3 \\
+ (-4m^2n - m^2 + 14mn - 12n + 3) \cdot N(D,n)^2 \\
+ (-15m^2n + 21mn - m - n - 4) \cdot N(D,n) - 9m^2n - 4m^2 + mn + 4m) = \\
(n \cdot N(D,n)^4 - (2mn - \frac{17}{2} n) \cdot N(D,n)^3 + (m^2n - 13mn + 11n + 2) \cdot N(D,n)^2 + \\
+ (6m^2n - 13mn - 4m + 5n + 2) \cdot N(D,n) + \left(\frac{7}{2} m^2n - \frac{7}{2} mn + 2m^2 - 2m\right) + \\
+ (mn + n) \cdot N(D,n)^4 + \left(-m^2n + 3mn + m + 3n + 1\right) \cdot N(D,n)^3 + \\
+ (-4m^2n - m^2 + 14mn - 12n + 3) \cdot N(D,n)^2 + \\
+ (-15m^2n + 21mn - m^2 - m - n - 4) \cdot N(D,n) - 9m^2n - 4m^2 + mn + 4m) = \\
(mn + 2n) \cdot N(D,n)^4 + (-m^2n + mn + m + \frac{23}{2} n + 1) \cdot N(D,n)^3 \\
+ (-3m^2n - m^2 + mn - n + 5) \cdot N(D,n)^2 \\
+ (-9m^2n + 8mn - m^2 - 5m + 4n - 2) \cdot N(D,n) - \frac{11}{2} m^2n - 2m^2 - \frac{5}{2} mn + 2m
\] (12)

(2) \( N(D,n) - m + 1 \leq i \leq W \)

\[
|B_{i-1}| = N(D,n) \\
T_i = (4n \cdot N(D,n) + 4n + 4) \cdot |B_{i-1}| + 4n = \\
= 4n \cdot N(D,n)^2 + (4n + 4) \cdot N(D,n) + 4n
\]
\[
\begin{align*}
\therefore \quad \sum_{i=N(D,n)-m+1}^{w} T_i &= |CP_{N(D,n)-m}| T_i = \\
&= \left[ \frac{(N(D,n)-m)(N(D,n)-m-1)}{2} + (m-1)(N(D,n)-m) + \frac{m(m-1)}{2} \right] \cdot (4n \cdot N(D,n)^2 + (4n+4) \cdot N(D,n) + 4n) = \\
&= \left[ \frac{1}{2} (N(D,n)^2 - 2m \cdot N(D,n) + m^2 - N(D,n) + m) + m \cdot N(D,n) - N(D,n) - m^2 + m \right] \cdot \frac{1}{2} - \frac{1}{2}m] = (4n \cdot N(D,n)^2 + (4n+4) \cdot N(D,n) + 4n) = \\
&= \left[ \frac{1}{2} N(D,n)^2 - \frac{3}{2} N(D,n) + m \right] \cdot \frac{1}{2} \cdot (4n \cdot N(D,n) + 3n) + (12) \cdot (13) = \\
&= (2m^2n - 2mn) \cdot N(D,n) + \frac{3}{2} m^2n - \frac{3}{2}mn + (mn + 2n) \cdot N(D,n) + \\
&\quad + (-m^2n + mn + m + \frac{23}{2} m + 1) \cdot N(D,n) + (-3m^2n - m^2 + mn - n + 5) \cdot N(D,n)^2 + \\
&\quad + (-9m^2n + 8mn - m^2 - 5m + 4n - 2) \cdot N(D,n) - \frac{11}{2} m^2n - 2m^2 - \frac{5}{2} mn + 2m + \\
&\quad + 2n \cdot N(D,n)^4 - (4n - 2) \cdot N(D,n)^3 + (4mn - 4n - 6) \cdot N(D,n)^3 + \\
&\quad + (4mn + 4m - 6n) \cdot N(D,n) + 4mn = \\
&= (mn + 4n) \cdot N(D,n)^4 + (-m^2n + mn + m + \frac{15}{2} n + 31) \cdot N(D,n)^3 + \\
&\quad + (-3m^2n - m^2 + 5mn - 5n - 1) \cdot N(D,n)^2 + \\
&\quad + (-7m^2n + 10mn - m^2 - m - 2n - 2) \cdot N(D,n) + \\
&\quad + 4m^2n - 2m^2 + 2m
\end{align*}
\]

So the complexity of S-polynomial reduction algorithm is
\[
T = (mn + 4n) \cdot N(D,n)^4 + \\
\quad + (-m^2n + mn + m + \frac{15}{2} n + 31) \cdot N(D,n)^3 + (-3m^2n - m^2 + 5mn - 5n - 1) \cdot N(D,n)^2 + \\
\quad + (-7m^2n + 10mn - m^2 - m - 2n - 2) \cdot N(D,n) + 4m^2n - 2m^2 + 2m
\]

This completes the proof. □

3.4 Complexity comparison and evaluation

When we put the complexities of above algorithms together and compare, then that is following.
Table 1. The complexity of S-polynomial reduction sub-algorithm

| algorithm                        | complexity                                                                 |
|----------------------------------|---------------------------------------------------------------------------|
| 1 F5-reduction algorithm of F5B  | $(2n \cdot N(D,n)^2 + (2n + 2) \cdot N(D,n)) \cdot |B_{i-1}|^2 +$            |
|                                  | $(4n \cdot N(D,n)^2 + 7n \cdot N(D,n)) \cdot |B_{i-1}| +$               |
|                                  | $(mn + n) \cdot N(D,n)^3 + (mn + m + 1) \cdot N(D,n)^2$                   |
| 2 new S-reduction algorithm      | $(2n \cdot N(D,n)^2 + 7n \cdot N(D,n)) \cdot |B_{i-1}| +$               |
|                                  | $(mn + n) \cdot N(D,n)^3 +$                                               |
|                                  | $(mn + m + 2n + 1) \cdot N(D,n)^2 +$                                      |
|                                  | $(n + 2) \cdot N(D,n)$                                                    |

As you see in the table 1, the complexity of F5-reduction algorithm of F5B is about $N(D,n)^2 \cdot |B_{i-1}|^2$ but the complexity of new S-reduction algorithm that we suggest is about $N(D,n)^2 \cdot |B_{i-1}|$, so our algorithm is very efficient because the deg of $|B_{i-1}|$ is lower as one deg.

As you see in the table 2, the complexity of Buchberger algorithm and F5B is about $n \cdot N(D,n)^5$ but the complexity of new algorithm that we suggest is about $mn \cdot N(D,n)^4$, so our algorithm is very efficient because the deg of $N(D,n)$ is lower as one deg.

Table 2. The complexity of algorithm for computing Gröbner basis

| algorithm           | complexity                                                                 |
|---------------------|---------------------------------------------------------------------------|
| 1 Buchberger algorithm | $\frac{3}{2} n \cdot N(D,n)^5 + N(D,n)^4 + 2mn \cdot N(D,n)^3$               |
|                     | $- m^2 n \cdot N(D,n)^2 - \frac{1}{2} m^2 n \cdot N(D,n)$                   |
| 2 F5B algorithm     | $\frac{2n}{3} N(D,n)^5 + mn \cdot N(D,n)^4 -$                                 |
|                     | $- m^2 n \cdot N(D,n)^3 + \frac{10}{3} m^3 n \cdot N(D,n)^2 -$              |
|                     | $- \frac{2}{3} m^3 n \cdot N(D,n) - \frac{11}{2} m^2 n$                     |
| 3 New algorithm     | $mn \cdot N(D,n)^4 - m^2 n \cdot N(D,n)^3 -$                                 |
|                     | $- 3m^2 n \cdot N(D,n)^2 - 7m^2 n \cdot N(D,n) +$                           |
|                     | $+ 4m^2 n$                                                                  |

**Attention:** we know that $m < N(D,n)$ for $m$ and $N(D,n)$.
4. CONCLUSIONS
We proposed a pair-selection-method newly to reduce S-polynomial quickly in the algorithms for computing Grobner basis.

5. FURTHER STUDY
In the future, we think it is very necessary to research the method for constructing Grobner bases like as mathematical programming. So we are going to make a lot of study to express the F4 algorithm already presented in [2] like as mathematical programming and combine the main idea of F5 with this in the future.

6. ACKNOWLEDGMENTS
Authors would like to thank anonymous reviewers’ help and advice.

7. REFERENCES
[1] Ars, G., J.-C. Faugère, H. Imai, M. Kawazoe and M. Sugita, Comparison between XL and Gröbner basis algorithms, in: P. J. Lee, editor, ASIACRYPT 2004, Lecture Notes in Computer Science 3329 (2004), pp. 338–353.
[2] Faugère, J.C., 1999. A new efficient algorithm for computing Gröbner bases (f4). Journal of Pure and Applied Algebra, 139(1), 61-88.
[3] Faugère, J.C., 2002. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). Symbolic and Algebraic Computation, Porc. Conferenz ISSAC 2002, 75-83.
[4] Faugère, J.C., 2010 FGb library for comptuing Gröbner bases. available at http://www-salsa.lip6.fr/jcf/Software/FGb/
[5] Faugère, J.C., September 03 – 06, 2013, F4 algorithm-F5 algorithm, GBRELA Workshop 2013
[6] Justin M. Gash.,2008. ON EFFICIENT COMPUTATION OF GRÖBNER BASES. Thesis for the degree of doctor of philosophy.
[7] Kenneth H. 2009, Rosen, DISCRETE MATHEMATICS AND ITS APPLICATIONS.
[8] Makoto S, Mitsuru K, Hideki I, 2004, Relation between XL algorithm and Gröbner Bases Algorithms. Cryptology ePrint Archive, Report 2004/112. http://eprint.iacr.org/
[9] Mora, T., M’oller, H.M. Traverso, C., 1992. Gröbner Bases Computation Using Syzygies. In ISSAC 92, P. S.Wang, Ed., ACM Press, Berkeley, California, United States, 320-328.
[10] Sun, Y. Wang, D.K., 2010. The F5 Algorithm in Buchberger’s Style, http://arxiv.org/abs/1006.5299v2
[11] Stegers, T., Faugère’s F5 algorithm revisited. Cryptology ePrint Archive, Report 2006/404, 2006. http://eprint.iacr.org/