Technical Reports on Mathematical and Computing Sciences: TR-C175
title: On the Degree of Multivariate Polynomials over Fields of Characteristic 2
author: Marcel Crasmaru [1]
affiliation:
1. Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology
Meguro-ku Ookayama, Tokyo 152-8552, Japan
(marcel@is.titech.ac.jp)

Abstract: We show that a problem of deciding whether a formula for a multivariate polynomial of n variables over a finite field of characteristic 2 has degree n when reduced modulo a certain Boolean ideal belongs to $\bigoplus$**P**. When the formula is allowed to have succinct representations as sums of monomials, the problem becomes $\bigoplus$**P**–complete.

1 Introduction

The class of languages accepted by unambiguous alternating Turing Machines, i.e. **UAP**, was introduced by Niedermeier and Rossmanith in [2]. Our object is to analyze the next decision problem:

`DEGREE-MOD-I`$_2$:
Instance: A formula for a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$, over a finite field $\mathbb{F}$ of characteristic 2.
Question: Does f have degree n when reduced modulo the Boolean ideal $\mathtt{I}_2 = \langle x_1^2 - x_1, \ldots, x_n^2 - x_n \rangle$?
Using the results of [3], it was shown in [4] that this problem is polynomial-time many-one hard for **UAP**.

The fact that **UAP** contains **FewP** (see [2]) makes `DEGREE-MOD-I`$_2$ problem computationally hard. But what is the exact complexity of `DEGREE-MOD-I`$_2$? One may observe that f reduces to 1 or 0 iff it is a tautology or it is unsatisfiable. It follows that the related question of whether the degree mod $\mathtt{I}_2$ is *positive* is **NP**–hard. However, it does not follow simply that our problem of whether the degree is n is **NP**–hard. The problem of deciding whether a given set of polynomials reduces to 1 under the *Gröbner basis* algorithm belongs to the second level of the polynomial hierarchy [5]. It is easy to check that $\{x_1^2 - x_1, \ldots, x_n^2 - x_n\}$ is a Gröbner basis for $\mathtt{I}_2$, but showing that `DEGREE-MOD-I`$_2$ belongs to the second level of the polynomial hierarchy would put **UAP** inside **PH** which is also questionable.

In the following, we shall show that that `DEGREE-MOD-I`$_2$ is decidable in $\bigoplus$**P**. Moreover, for a slightly general class of formulas, i.e., formulas allowing succinct representations of sums of monomials, we show that the corresponding `DEGREE-MOD-I`$_2$ problem is $\bigoplus$**P**–complete. We also introduce a generalized problem, called `Non-Reducible-Monomial`, and prove that it is $\bigoplus$**P**—complete.

2 Preliminaries

Let $\mathbb{F}$ be a finite field of characteristic 2 and let $\mathbf{R} = \mathbb{F}[x_1, \ldots, x_n]$ be the ring of polynomials with coefficients in $\mathbb{F}$ and variables $x_1, \ldots, x_n$.

In this paper our model of computation is the Turing machine model. We assume as given a suitable binary encoding for the elements of $\mathbb{F}$, and also for the monomials of ring $\mathbf{R}$. We also assume that the addition and the multiplication in $\mathbb{F}$ can be done in a unit of time. The sum and the multiplication of two monomials in $\mathbf{R}$ are taken as granted, implemented as oracles. More precisely, there are two oracle like Turing machines, which take as input the encoding of two monomials and return the sum and the multiplication of the input monomials, respectively.

Let $\mathcal{M}(\mathbf{R}, \mathtt{I}_2)$ be the set of monomials of the factor ring $\mathbf{R}/\mathtt{I}$. As an example, $\mathcal{M}(\mathbf{R}, \mathtt{I}_2) = \{0, 1, x_1, x_2, x_1 \cdot x_2\}$, when $\mathbf{R} = \mathbb{F}[x_1, x_2]$, and $\mathtt{I}_2 = \langle x_1^2 - x_1, x_2^2 - x_2 \rangle$.

Let $\mathbf{R} = \mathbb{F}[x_1, \ldots, x_n]$. We would like to generalize expressions for formulas by introducing Σ-expressions. First, we give a syntactic definition. A Σ-*formula* is defined inductively as follows.

Definition 2.1 *1. Any monomial $m \in \mathbf{R}$ is a Σ-formula.*

2. If S is a multi-set of monomials of $\mathbf{R}$, then $\sum_{m \in S} m$ is a Σ-formula.

3. If ϕ and ψ are Σ-formulas, then $\phi + \psi$ is a Σ-formula.

4. If m and m' are monomials, then $m \cdot m'$ is a Σ-formula.

5. If m is a monomial and ψ is a Σ-formula, then $m \cdot (\psi)$ is a Σ-formula.

6. If ϕ and ψ are Σ-formulas, then $(\phi) \cdot (\psi)$ is a Σ-formula.

Note that, for specifying a Σ-formula such as $f = (\Sigma_{m \in S_1} m + x_1 x_5)(\Sigma_{m \in S_2} m + x_3^2 x_7 x_8 + \Sigma_{m \in S_3} m)$, we need to specify S_1, S_2, and S_3. In the sequel we specify a multi-set of monomials $S \subseteq \mathbf{R}$ by giving a Boolean circuit C_S. Loosely speaking, C_S takes a binary representation of a monomial m and decides whether m belongs to S. However, more consideration is necessary for taking care of multi-sets. More precisely, elements of S are indexed according to some set $\mathcal{I}$, and the circuit C_S, given a pair of an index i and a monomial m, decides whether m has the index i in S. We assume that an input pair (i, m) is encoded as a binary string of a certain length ℓ that is the input size of C_S. Clearly, S contains no monomial that can not be encoded within this length ℓ. For specifying a Σ-formula f as input, we shall assume that representation of such circuits are given along with the expression of f.

Suppose, for example, $f = (\Sigma_{m \in S} + \cdots)(\cdots)$ is given as an input formula. Clearly it is polynomial-time (w.r.t. the representation of f) to decide, for a given (i, m), whether m has index i in S, where S was apriori indexed according to some set $\mathcal{I}$. Using this property, it is also possible to guess all and only elements of S in nondeterministic polynomial time w.r.t. the formula representation. Note that S may contain *exponentially* many elements; thus, nondeterminism is needed in general.

Now we are ready to define a generalized version of $\mathtt{DEGREE\text{-}MOD\text{-}I}_2$ problem.

Σ-$\mathtt{DEGREE\text{-}MOD\text{-}I}_2$:
Instance: A Σ-formula for a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$, over a finite field $\mathbb{F}$ of characteristic 2.
Question: Does f have degree n when reduced modulo the Boolean ideal $\mathtt{I}_2 = \langle x_1^2 - x_1, \ldots, x_n^2 - x_n \rangle$?

We also consider another generalization of `DEGREE-MOD-I`$_2$, which is defined as follows:

`NON-RED-MONOMIAL`$(f, m, \mathtt{I})$:
INSTANCE: A Σ-formula f for a polynomial and a monomial m in $\mathbf{R} = \mathbb{F}[x_1, ..., x_n]$. Also a description of an ideal $\mathtt{I}$ in $\mathbf{R}$, i.e., a list of polynomials $\{p_1, ..., p_m\}$ such that $\mathtt{I} = \langle p_1, ..., p_m \rangle$ and a deterministic Turing machine M that decides whether a given pair of monomials $(m', m'') \in \mathbf{R}^2$ verifies $m' \equiv m'' \pmod{\mathtt{I}}$ within time polynomial in the length of given monomials.
QUESTION: Does f contain the monomial m when reduced modulo the ideal $\mathtt{I}$?

The description of ideal $\mathtt{I}$ stated in the definition of `NON-RED-MONOMIAL` problem is necessary. Otherwise, to decide whether two monomials are equivalent modulo $\mathtt{I}$ reduces to an instance of ideal membership problem, known to be an intractable problem (see [6]). Again it is easy to see that `DEGREE-MOD-I`$_2$ is polynomial-time reducible to `NON-RED-MONOMIAL`$(f, x_1 x_2 \cdots x_n, \mathtt{I}_2)$.

In the next section we present a $\bigoplus\mathbf{P}$ algorithm that decides whether a Σ-formula contains a certain non-reducible monomial.

3 A $\bigoplus\mathbf{P}$ Algorithm Solving `NON-RED-MONOMIAL`$(f, m, \mathtt{I})$

Let $\mathtt{I}$ be an ideal in $\mathbf{R} = \mathbb{F}[x_1, ..., x_n]$, such that $\mathtt{I}$ verifies the hypotheses of `NON-RED-MONOMIAL` problem. Let us consider the following nondeterministic algorithm, which takes as input a Σ-formula f in $\mathbf{R}$, and two monomials $m, m' \in R$ and outputs *accept* or *reject*.

```
procedure findMonomial(Formula f, Monomial m, Monomial m') {
 if(f = 0) {reject; halt;}
 if(f = 1) {
  if(m ≡ m' (mod I)) {
    accept;
    halt;
  } else {
    reject;
    halt;
  }
 }
 if(f = m'' and m'' is Monomial ) {
  call findMonomial(1, m, m'·m'');
 }
 if(f = ∑_{p∈S}) {        // where S is given by Boolean circuit C_S
  nondeterministically guess m'' ∈ S;
  call findMonomial(m'', m, m');
 }
 if(f = a_1 + a_2 + ··· + a_k) {
  nondeterministically guess h ∈ {a_1, ..., a_k};
  call findMonomial(h, m, m');
 }
```

21 if($f = m'' \cdot b$) {
22   call findMonomial(b, m, $m' \cdot m''$);
23 }
24 if($f = (\sum_{p \in S} p) \cdot b$) {
25   nondeterministically guess $h \in S$;
26   call findMonomial($h \cdot b$, d, m);
27 }
28 if($f = (a_1 + a_2 + \cdots + a_k) \cdot b$) {
29   nondeterministically guess $h \in \{a_1, \ldots, a_k\}$;
30   call findMonomial($h \cdot b$, d, m);
31 }
} // *end procedure*

Notice that by the way of specifying the multi-set S, the line 14 of the algorithm contains a polynomial nondeterministic guess. That is, one may verify in polynomial-time that the guessed monomial m'' belongs to S. This fact and the hypothesis that deciding whether two monomials are equivalent modulo I takes time polynomial in the length of given monomials assure us that our algorithm may be implemented by a nondeterministic Turing machine running in polynomial time w.r.t. the length of the input.

Let $\widehat{f}$ be the formula obtained from f using the distributive law of the ring $\mathbf{R}$, but without collecting identical terms. For example, if $f = (x_1+x_2)(x_1+x_2)$, then $\widehat{f} = x_1^2+x_1\cdot x_2+x_1\cdot x_2+x_2^2$.

Lemma 3.1 *The algorithm* `findMonomial`*(f, m, q) finishes in accept state iff there exists at least one monomial m' in $\widehat{f}$ such that $m \equiv m' \cdot q \pmod{\mathtt{I}}$.*

Proof: Let us check the statement of the lemma in some different basic cases. Given two monomials $f, m \in \mathcal{M}(\mathbb{F}, \mathtt{I})$, one can observe that findMonomial(f, m, q) returns accept iff findMonomial(1, m, $q \cdot f$) returns accept. It follows that $m \equiv f \cdot q \pmod{\mathtt{I}}$.

Consider now $f = \sum_{p \in S}$, where S is a multi-set of monomials. Let m be a monomial. Notice that findMonomial(f, m, q) returns accept iff there exists $m' \in S$ such that findMonomial(m', m, q) returns accept (lines 14 and 15 in the above algorithm). Then it follows that there exists $m' \in S$ such that $m \equiv m' \cdot q \pmod{\mathtt{I}}$.

Suppose that $f = \psi + \phi$, and findMonomial(g, m, q) returns accept iff there is at least one monomial m' in $\widehat{g}$ such that $m \equiv m' \cdot q \pmod{\mathtt{I}}$, for some $g \in \{\psi, \phi\}$ and all $m, q \in \mathcal{M}(\mathbb{F}, \mathtt{I})$. Then we see that findMonomial(f, m, q) returns accept iff there exists h such that $\psi = \alpha + h + \beta$, or $\phi = \gamma + h + \delta$ and findMonomial(h, m, q) returns accept (lines $17 - 19$ in the above algorithm). It follows that findMonomial(ψ, m, q) returns accept or findMonomial(ϕ, m, q) returns accept iff there is at least one monomial m' such that $m \equiv m' \cdot q \pmod{\mathtt{I}}$ and $m' \in \widehat{\psi}$ or $m' \in \widehat{\phi}$ (our hypothesis). We conclude that there exists at least one monomial m' in $\widehat{f}$ that verifies the condition $m \equiv m' \cdot q \pmod{\mathtt{I}}$.

Suppose that $f = p \cdot \psi$, where p is a monomial and findMonomial(ψ, m, q) returns accept iff there is at least one monomial m' in $\widehat{\psi}$, such that $m \equiv m' \cdot q \pmod{\mathtt{I}}$, for all $m \in \mathcal{M}(\mathbb{F}, \mathtt{I})$. By inspecting the algorithm it follows that findMonomial(f, m, q) returns accept iff findMonomial(ψ, m, $q \cdot p$) returns accept. That is, there is at least one monomial m' in $\widehat{\psi}$, such that $m \equiv m' \cdot q \cdot p \pmod{\mathtt{I}}$. Observing that $\widehat{f}$ is obtained from $p \cdot \widehat{\psi}$, it follows that `findMonomial(`$f,\ m,\ q$`)` returns accept iff there exists at least one monomial $m'' = m' \cdot p$ in $\widehat{f}$, such that $m \equiv m'' \cdot q \pmod{\mathtt{I}}$.

Suppose that $f = (\sum_{p \in S} p) \cdot \psi$, where `findMonomial(`$\psi,\ m,\ q$`)` accepts iff there is at least one monomial m' in $\widehat{\psi}$, such that $m \equiv m' \cdot q \pmod{\mathtt{I}}$, for all $m \in \mathcal{M}(\mathbb{F}, \mathtt{I})$. Then the lines $25 - 26$ of the algorithm show us that `findMonomial(`$f,\ m,\ q$`)` returns accept iff there exists a monomial $u \in S$ such that `findMonomial(`$u \cdot \psi,\ m,\ q$`)` returns accept. This is equivalent to `findMonomial(`$\psi,\ m,\ q \cdot u$`)` returns accept. We conclude that there is at least one monomial m' in $\widehat{\psi}$, such that $m \equiv m' \cdot q \cdot u \pmod{\mathtt{I}}$. Because $\widehat{f}$ is obtained from $(\sum_{p \in S} p) \cdot \widehat{\psi}$, it follows that `findMonomial(`$f,\ m,\ q$`)` accepts iff there is at least one monomial $m'' = u \cdot m'$ in $\widehat{f}$, such that $m \equiv m'' \cdot q \pmod{\mathtt{I}}$.

Then the proof follows by structural induction on f, by observing that the other remaining cases reduce easily to the above proved ones. □

Lemma 3.2 *The number of accepting paths of the nondeterministic algorithm* `findMonomial(`$f,\ m,\ 1$`)` *is equal to the number of monomials m' of $\widehat{f}$ verifying the condition* $m \equiv m' \pmod{\mathtt{I}}$.

Proof: The proof follows easily by structural induction on f, using Lemma 3.1. □

Theorem 3.3 *The problem NON-RED-MONOMIAL$(f, m, \mathtt{I})$ belongs to $\bigoplus\mathbf{P}$.*

Proof: Let M be a NTM that takes as input a Σ-formula f and a monomial m and computes `findMonomial(`$f,\ m,\ 1$`)`. Such a machine can be easily designed using our algorithm.

By Lemma 3.2, we know that the number of accepting paths of M when the input is f and m (denoted $\#M(f, m)$), is precisely the number of monomials m' of $\widehat{f}$ verifying the condition $m \equiv m' \pmod{\mathtt{I}}$. The field $\mathbb{F}$ being of characteristic 2 it follows that $\#M(f, m) \equiv 1 \pmod 2$ iff the formula f contains the monomial m when reduced modulo the ideal $\mathtt{I}$. □

One can observe that in the above presented algorithm, by modifying line 2 from

`if(`$m \equiv m' \pmod{\mathtt{I}}$`) {`

to `if(`$m = m'$`) {`,

we obtain a $\bigoplus\mathbf{P}$ algorithm for deciding whether a polynomial given as a Σ-formula contains a certain monomial. By further commenting out lines $13 - 16$, and $24 - 27$, we have a $\bigoplus$-$\mathbf{P}$ algorithm for deciding whether a (normal) formula reduces to a polynomial containing a certain monomial.

Corollary 3.4 *Given a formula for a polynomial $p \in \mathbf{R}$ and a monomial $m \in \mathbf{R}$, one may decide in $\bigoplus\mathbf{P}$ whether p contains m.*

4 Σ-DEGREE-MOD-$\mathtt{I}_2$ is $\bigoplus\mathbf{P}$–Complete

It is well known that $\bigoplus$HAMILTON PATH problem is $\bigoplus\mathbf{P}$–complete (see, for example, [1].)

INSTANCE: A digraph $G = (V, E)$.

QUESTION: Is the number of Hamilton paths of G odd?

We will show that $\bigoplus$HAMILTON PATH reduces polynomial-time many to one to Σ-DEGREE-MOD-$\mathtt{I}_2$. Let $G = (V, E)$ be a graph having n vertices and edges $E \subseteq V^2$.

We continue to assume that $\mathbb{F}$ is a finite field of characteristic 2, and let $\mathbf{R} = \mathbb{F}[x_1, \ldots, x_n]$ be the ring of polynomials with coefficients in $\mathbb{F}$ and variables $x_1, \ldots, x_n$. Let $\mathcal{M}(\mathbb{F}, \mathtt{I}_2)$ be the set of monomials of the ring $\mathbf{R}/\mathtt{I}_2$, where $\mathtt{I}_2$ is the ideal $\langle x_1^2 - x_1, \ldots, x_n^2 - x_n \rangle$. We label vertices of G such that $V = \{x_1, x_2, \ldots, x_n\}$ and consequently identify the edges of E correspondingly. Let S_G be the following multi-set of monomials:

$$S_G = \{v_1 \cdot v_2 \cdots v_n \mid (v_i, v_{i+1}) \in E, \text{when } 1 \le i \le n-1\}.$$

Let C_{S_G} be the Boolean circuit defined as follows. We assume that the variables $x_1, \ldots, x_n$ have all the same length when binary encoded. The input of C_{S_G} is a pair (i, m), where i is the binary encoding of an element of the set $\{x_1, x_2, \ldots, x_n\}^n$ and m is the binary encoding of a monomial. A requirement for m is that the length of its binary encoding to be equal to the length of binary encoding of monomial $x_1 x_2 \cdots x_n$, some padding being allowed. This assures us that the input of the circuit C_{S_G} has constant length. The circuit returns `true` if and only if $m = i_1 \cdot i_2 \cdots i_n$ and $i_1, i_2, \ldots, i_n$ is a *walk* in G, i.e. $(i_j, i_{j+1}) \in E$ for $1 \le j \le n-1$, when $i = (i_1, i_2, \ldots, i_n)$. Such a circuit can be easily designed as follows. First we construct a circuit that checks whether $m = i_1 \cdot i_2 \cdots i_n$, this problem being obviously implementable by a Boolean circuit. Second we design a circuit that verifies whether $i_1, i_2, \ldots, i_n$ is a walk in G. It is a known result that such a Boolean circuit can be implemented. Finally, C_{S_G} is the logical `AND` of the two already constructed circuits. It follows that C_{S_G} accepts (i, m) iff m is an element of index i in S_G, where S_G is indexed by the set of walks of length n in G. Then it follows that formula

$$f_G = \sum_{m \in S_G} m$$

is a Σ-formula.

Next we analyze the problem Σ-DEGREE-MOD-$\mathtt{I}_2$ for f_G that is clearly equivalent to NON-RED-MONOMIAL$(f_G, x_1 x_2 \cdots x_n, \mathtt{I}_2)$.

Lemma 4.1 *The number of Hamilton paths of $G = (V, E)$ is odd iff f_G contains the monomial $x_1 x_2 \cdots x_n$ when reduced modulo $\mathtt{I}_2$.*

Proof: Let $v_1, v_2, \ldots, v_n$ be a Hamilton path of G. By definition, we have that $(v_i, v_{i+1}) \in E$ for all $1 \le i \le n-1$, and $\{v_1, v_2, \ldots, v_n\} = \{x_1, x_2, \ldots, x_n\}$. It follows that $v_1 \cdot v_2 \cdots v_n$ belongs to the multi-set $T_G = \{m \in S_G \mid m \equiv x_1 x_2 \cdots x_n \pmod{\mathtt{I}_2}\}$, that is, the multi-set of the monomials m belonging to S_G that verify the property $m \equiv x_1 x_2 \cdots x_n \pmod{\mathtt{I}_2}$.

Consider then the map $H : \{[v_1, v_2, \ldots, v_n] \in V^n \mid v_1, v_2, \ldots, v_n \text{ Hamilton path of } G\} \longrightarrow T_G$, given by

$$H([v_1, v_2, \ldots, v_n]) = v_1 \cdot\ v_2 \cdots\ v_n.$$

From the definition of T_G one can observe that for any $v_1 \cdot v_2 \cdots v_n \in T_G$, we have that $H([v_1, v_2, \ldots, v_n]) = v_1 \cdot v_2\ \cdots\ v_n$. This shows that H is surjective. For any two Hamilton paths p_1 and p_2 such that $p_1 \neq p_2$, it is easy to check that their images $H(p_1)$ and $H(p_2)$ verify $H(p_1) \neq H(p_2)$. It follows that H is bijective.

Then it follows that the number of Hamilton paths of G is odd iff $|T_G|$ is odd iff f_G contains monomial $x_1 x_2 \cdots x_n$ when reduced modulo $\mathtt{I}_2$. $\square$

Finally, from Lemma 4.1 and Theorem 3.3 we deduce the following theorem.

Theorem 4.2 *Σ-DEGREE-MOD-$\mathtt{I}_2$ is $\bigoplus$**P**—complete.*

A straightforward corollary follows.

Corollary 4.3 *NON-RED-MONOMIAL$(f, m, \mathtt{I})$ is $\bigoplus$**P**–hard.*

5 NON-RED-MONOMIAL$(f, m, \mathtt{I})$ Remains $\bigoplus$P–Hard When Restricted to Normal Formulas

One can observe that our proof that NON-RED-MONOMIAL$(f, m, \mathtt{I})$ is $\bigoplus$**P**—hard uses the fact that a Σ-formula may contain succinct representations of (possibly) exponential large sums of monomials. Here we show that one can restrict NON-RED-MONOMIAL$(f, m, \mathtt{I})$ to "normal" formulas, but NON-RED-MONOMIAL$(f, m, \mathtt{I})$ still remains $\bigoplus$**P**—complete for some ideals.

In what follows, a formula is given by modifying definition 2.1 such that sums over multi-sets of monomials are not allowed (previous definitions remain granted).

Lemma 5.1 *The decision problem NON-RED-MONOMIAL$(f, m, \mathtt{I})$ restricted to normal formulas belongs to $\bigoplus$**P**.*

Proof:

The proof follows as a simplified version of the proofs of Lemmas 3.1 and 3.2, by using instead a new program obtained from the algorithm presented in Section 3 by commenting out the lines $13 - 16$ and $24 - 27$. $\square$

Lemma 5.2 *The decision problem NON-RED-MONOMIAL$(f, m, \mathtt{I})$ restricted to normal formulas is $\bigoplus$**P**—hard.*

Proof: We shall reduce $\bigoplus$ HAMILTON PATH to NON-RED-MONOMIAL$(f, x_1 x_2 \cdots x_n, \mathtt{I}_{nt})$, with f and $\mathtt{I}_{nt}$ defined as follows.

Let $G = (V, E)$ be a digraph. Without lose of generality we may assume that there are precisely two vertices $a, b \in V$ such that $out(a) = V - \{a, b\}$, $in(b) = V - \{a, b\}$, and $in(a) = out(b) = \emptyset$. If such vertices fail to exist, we just add them to V and also add the necessary edges to E. It is easy to see that the number of Hamilton paths is invariant to the above transformation.

Let $|V| = n$ and $|E| = t$. Let $\mathbb{F}$ be a finite field of characteristic 2, and let let $\mathtt{I}_{nt} = \langle y_1 \cdot z_1 - 1, \ldots, y_t \cdot z_t - 1 \rangle$ be an ideal in the ring $\mathbf{R} = \mathbb{F}[x_1, \ldots, x_n, y_1, \ldots, y_t, z_1, \ldots, z_t]$.

We may assume that the vertices in V are labelled by $x_1, \ldots, x_n$ such that a is labelled by x_1 and b is labelled by x_n. Then we label the set of edges E by the set of pairs $P = \{(y_1, z_1), (y_2, z_2), \ldots, (y_t, z_t)\}$. In what follows we identify the vertices and the edges by their labels.

Consider the following formula.

$$f_G = \Big(\sum_{(y_{j_1},z_{j_1})\in out(x_1)} x_1 \cdot y_{j_1}\Big)\Big(\sum_{\substack{[(y_{i_2},z_{i_2}),(y_{j_2},z_{j_2})]\\ \in in(x_2)\times out(x_2)}} z_{i_2}\cdot x_2 \cdot y_{j_2}\Big)\cdot\Big(\sum_{\substack{[(y_{i_k},z_{i_k}),(y_{j_k},z_{j_k})]\\ \in in(x_k)\times out(x_k)}} z_{i_k}\cdot x_k\cdot y_{j_k}\Big)\cdots\Big(\sum_{(y_{i_n},z_{i_n})\in in(x_n)} z_{i_k}\cdot x_n\Big).$$

Let S_G be the multi-set of monomials of $\widehat{f_G}$ and define the multi-set

$$T_G = \{m \in S_G \mid m \equiv x_1x_2\cdots x_n \pmod{\mathtt{I}_{nt}}\}.$$

Let $p = v_1, (y_{12}, z_{12}), v_2, \dots, (y_{n-1n}, z_{n-1n}), v_n$ be a Hamilton path of G. By definition, we have that $(v_i, v_{i+1}) = (y_{ii+1}, z_{ii+1}) \in E$, for all $1 \le i \le n-1$ and $\{v_1, v_2, \dots, v_n\} = \{x_1, x_2, \dots, x_n\}$. Any Hamilton path starts from a and finishes at b, whence $v_1 = x_1$ and $v_n = x_n$. Then $(v_1 \cdot y_{12})(z_{12}v_2y_{23})\cdots(z_{nn-1}v_n) \in S_G$ because it has to appear in $\widehat{f_G}$. The fact that $(v_1 \cdot y_{12})(z_{12}v_2y_{23})\cdots(z_{nn-1}v_n) \equiv x_1x_2\cdots x_n \pmod{\mathtt{I}_{nt}}$ shows us that $(v_1{\cdot}y_{12})(z_{12}v_2y_{23})\cdots(z_{nn-1}v_n) \in T_G$.

Consider then the map $H : \{[v_1, (y_{12}, z_{12}), v_2, \dots, (y_{n-1n}, z_{n-1n}), v_n] \in V \times E \times \cdots \times E \times V \mid v_1, (y_{12}, z_{12}), v_2, \dots, (y_{n-1n}, z_{n-1n}), v_n \text{ Hamilton path of } G\} \longrightarrow T_G$, given by

$$H\big([v_1, (y_{12}, z_{12}), v_2, \dots, (y_{n-1n}, z_{n-1n}), v_n]\big) = (v_1 \cdot y_{12})(z_{12}v_2y_{23})\cdots(z_{nn-1}v_n).$$

By the definition of T_G, one can observe that for any $h = v_1{\cdot}(y_{12}z_{12}){\cdot}v_2{\cdot}(y_{23}z_{23})\cdots(y_{n-1n}z_{n-1n}){\cdot} v_n \in T_G$, we have that $H([v_1, (y_{12}, z_{12}), v_2, \dots, (y_{n-1n}, z_{n-1n}), v_n]) = h$. Thus H is surjective. For any two Hamilton paths p_1 and p_2 such that $p_1 \ne p_2$, it is easy to check that their images $H(p_1)$ and $H(p_2)$ verify $H(p_1) \ne H(p_2)$, recalling that T_G is multi-set. It follows that H is bijective.

This shows that the number of Hamilton paths of G is odd iff $|T_G|$ is odd iff f_G contains monomial $x_1x_2\cdots x_n$ when reduced modulo $\mathtt{I}_{nt}$ and completes the proof of the lemma. □

6 The Complexity of DEGREE-MOD-$\mathtt{I}_2$ and Conclusions

Because DEGREE-MOD-$\mathtt{I}_2$ is a special instance of NON-RED-MONOMIAL$(f, m, \mathtt{I})$ it follows that DEGREE-MOD-$\mathtt{I}_2$ is decidable in $\bigoplus$**P**. We know that DEGREE-MOD-$\mathtt{I}_2$ is **UAP**—hard. Unfortunately we can not prove that DEGREE-MOD-$\mathtt{I}_2$ is complete for **UAP** or $\bigoplus$**P**. What we can prove, however, is that the problem that corresponds to DEGREE-MOD-$\mathtt{I}_2$ for Σ-formulas, i.e. Σ-DEGREE-MOD-$\mathtt{I}_2$, is $\bigoplus$**P**—complete.

It still remains an open issue to precisely locate DEGREE-MOD-$\mathtt{I}_2$ problem between **UAP** and $\bigoplus$**P**. A nice result would be to prove that DEGREE-MOD-$\mathtt{I}_2$ is **UAP**–complete, by contrast to our results presented here. However, we have fully settled the complexity of a connected problem, by proving that NON-RED-MONOMIAL$(f, m, \mathtt{I})$ is $\bigoplus$**P**–complete.

Acknowledgement

I would like to thank Prof. Osamu Watanabe for many helpful suggestion for improving this paper.

References

[1] C.H. Papadimitriou, Computational complexity, Addison-Wesley,1994.

[2] R.Niedermeyer and R. Rossmanith, Unambiguos computions and locally definable acceptance types, *Theoretical Computer Science*, 194(1-2):137-161 1998.

[3] S.Aida, M.Crasmaru., K.Regan, and O.Watanabe, Games with a uniqueness property, STACS2002, *Lecture Notes in Computer Science*, 2002.

[4] M.Crasmaru., C.Glaßer, K.Regan, S.Sengupta, On global unique alternations, submitted, 2002.

[5] Pascal Koiran, Hilbert's Nullstellensatz is in the Polynomial ierarchy, *Journal of Complexity*, 12(4):273-286, 1996.

[6] E. Mayr and A. Meyer, The complexity of the word problems for commutative semigroups and polynomial ideals, *Adv. Math.*, 46:305-329, 1982.