Communications of JSSAC (2012)
Vol. 1, pp. 27 – 37

Efficient Singleton Set Constraint Solving by Boolean Gröbner Bases

Shutaro Inoue

Tokyo University of Science

(Received 30/Oct/2009 Accepted 29/Nov/2009)

Abstract

A set constraint with an additional restriction for each variable to be a singleton set is called a singleton set constraint. Many combinatorial problems such as a Sudoku puzzle can be considered as singleton set constraints. In this paper we introduce a new algorithm which solves singleton set constraints using only computations of Boolean Gröbner bases.

1 Introduction

Combinatorial problems are often reduced to solving certain types of constraints of set. Such set constraints are described as polynomial equations over Boolean rings. *Boolean Gröbner bases* introduced in [6] are powerful tools for solving such constraints and employed as a central engine of *Set Constraint Solver* [7, 9]. When a set constraint has an additional restriction on the cardinalities of variables, however, we can not solve it by simply computing the corresponding Boolean Gröbner basis. We need some additional computations in general.

A set constraint is called a *singleton set constraint* if we have a restriction that each variable is a singleton set. Many combinatorial problems such as a Sudoku puzzle is actually described as a singleton set constraint. In this paper we show a certain type of polynomials of an ideal of a Boolean polynomial ring can be obtained by computing only one Boolean Gröbner basis. The result leads us to have an efficient algorithm to solve singleton set constraints using only computations of Boolean Gröbner bases. The algorithm is implemented in [1] and its effectiveness is verified through our computation experiments.

Our paper is organized as follows. In section 2, we first show some classical results of Boolean algebra. In section 3, we give a quick review of Boolean Gröbner bases. Section 4 is devoted to our main result together with a new method to solve singleton set constraints. In section 5, we see our method is quite efficient for solving Sudoku puzzles by showing timing data we have obtained through our computation experiments.

2 Boolean polynomial ring

In this section, We give several definitions and notations about Boolean polynomial rings, then we show important classical results of Boolean algebra. More details with proofs can be found in [11].

Definition 1
A commutative ring $\mathbf{B}$ *with an identity* 1 *is called* a Boolean ring *if every element* a *of* $\mathbf{B}$ *is idempotent, i.e.* $a^2 = a$.

$(\mathbf{B}, \vee, \wedge, \neg)$ becomes a Boolean algebra with the Boolean operations $\vee, \wedge, \neg$ defined by $a \vee b = a + b + a \cdot b, a \wedge b = a \cdot b, \neg a = 1 + a$. Conversely, for a Boolean algebra $(\mathbf{B}, \vee, \wedge, \neg)$, if we define $+$ and $\cdot$ by $a + b = (\neg a \wedge b) \vee (a \wedge \neg b)$ and $a \cdot b = a \wedge b$, $(\mathbf{B}, +, \cdot)$ becomes a Boolean ring.
Since $-a = a$ in a Boolean ring, we do not need to use the symbol '$-$', however, we also use $-$ when we want to stress its meaning.

Example 2
Let S *be an arbitrary set and* $\mathcal{P}(S)$ *be its power set, i.e. the family of all subsets of* S*. Then,* $(\mathcal{P}(S), \vee, \wedge, \neg)$ *becomes a Boolean algebra with the operations* $\vee, \wedge, \neg$ *as the union, intersection and the complement of* S *respectively. As a Boolean ring, it is isomorphic to* $\mathbb{GF}_2^S$ *that is a commutative ring of all functions from* S *to* $\mathbb{GF}_2$*. Stone's representation theorem tells us any Boolean ring is isomorphic to sub-algebra of* $\mathbb{GF}_2^S$ *for some set* S*. Especially, when* $\mathbf{B}$ *is finite Boolean ring, it is isomorphic to a direct product* $\mathbb{GF}_2^k$ *for some natural number* k*. Note that a computable Boolean ring need not be finite. For any countable set* S*, any family of computable subsets* S *which is closed under the computable operations* $\vee, \wedge, \neg$ *is a computable Boolean ring.*

Definition 3
Let $\mathbf{B}$ *be a Boolean ring. A quotient ring* $\mathbf{B}[X_1, \ldots, X_n]/\langle X_1^2 - X_1, \ldots, X_n^2 - X_n\rangle$ *modulo an ideal* $\langle X_1^2 - X_1, \ldots, X_n^2 - X_n\rangle$ *becomes a Boolean ring. It is called* a Boolean polynomial ring *and denoted by* $\mathbf{B}(X_1, \ldots, X_n)$*, its element is called* a Boolean polynomial.

Note that a Boolean polynomial of $\mathbf{B}(X_1, \ldots, X_n)$ is uniquely represented by a polynomial of $\mathbf{B}[X_1, \ldots, X_n]$ that has at most degree 1 for each variable X_i. In what follows, we identify a Boolean polynomial with such a representation.

Multiple variables such as $A_1, \ldots, A_m$ or $X_1, \ldots, X_n$ are abbreviated to $\bar{A}$ or $\bar{X}$ respectively. Lower small roman letters such as a, b, c are usually used for elements of a Boolean ring $\mathbf{B}$. The symbol $\bar{a}$ denotes an m-tuple of element of $\mathbf{B}$ for some m. For a Boolean polynomial $f(\bar{A}, \bar{X})$ with variables $\bar{A}$ and $\bar{X}$, $f(\bar{a}, \bar{X})$ denotes a Boolean polynomial in $\mathbf{B}(\bar{X})$ obtained by specializing $\bar{A}$ with $\bar{a}$.

Definition 4
Let I *be an ideal of* $\mathbf{B}(X_1, \ldots, X_n)$*. For a subset* S *of* $\mathbf{B}$*,* $V_S(I)$ *denotes a subset* $\{\bar{a} \in S^n | \forall f \in I f(\bar{a}) = 0\}$*. When* $S = \mathbf{B}$*,* $V_{\mathbf{B}}(I)$ *is simply denoted by* $V(I)$ *and called* a variety *of* I*. We say* I *is* satisfiable *in* S *if* $V_S(I)$ *is not empty. When* $S = \mathbf{B}$*, we simply say* I *is* satisfiable.

Theorem 5 (Boolean extension theorem)
Let I *be a finitely generated*
ideal in a boolean polynomial ring $\mathbf{B}(Y_1, \ldots, Y_m, X_1, \ldots, X_n)$*.*
For any $\bar{b} \in V(I \cap \mathbf{B}(\bar{Y}))$*, there exist* $\bar{c} \in \mathbf{B}^n$ *such that* $(\bar{b}, \bar{c}) \in V(I)$*.*

Corollary 6 (Boolean weak Nullstellensatz)
For any finitely generated
ideal I *of a boolean polynomial ring* $\mathbf{B}(X_1, \ldots, X_n)$*, the variety* $V(I)(\subseteq \mathbf{B}^n)$ *of* I *is an empty set if and only if there exists a non-zero constant element of* $\mathbf{B}$ *in* I*.*

Theorem 7 (Boolean strong Nullstellensatz)
Let I be a finitely generated ideal of a boolean polynomial ring $\mathbf{B}(X_1, \dots, X_n)$ such that $V(I) \neq \emptyset$. Then, for any boolean polynomial $h(\bar{X}) \in \mathbf{B}(\bar{X})$,

$$h(\bar{X}) \in I \quad \textit{if and only if} \quad \forall(\bar{b}) \in V(I) \;\; h(\bar{b}) = 0.$$

3 Boolean Gröbner bases

A Boolean Gröbner basis is defined as a natural modification of a Gröbner basis in a Boolean polynomial ring. Though it was introduced in [5, 6] together with a computation algorithm using a special monomial reduction, the same notion was independently discovered in [12] for a polynomial ring over a more general coefficient ring. In this section, we give a quick review of Boolean Gröbner bases. More detailed descriptions with proofs can be found in [8] or [12].
In what follows, we assume that some term order on a set of power products of variables is given. For a polynomial f in a polynomial ring $\mathbf{B}[\bar{X}]$ over a Boolean ring $\mathbf{B}$, we use the notations $LT(f)$, $LM(f)$ and $LC(f)$ to denote the leading power product, the leading monomial and leading coefficient of f respectively. $f - LM(f)$ is also denoted by $Rd(f)$. We also use the notations $LT(F)$ and $LM(F)$ to denote the sets $\{LT(f)|f \in F\}$ and $\{LM(f)|f \in F\}$ for a (possibly infinite) subset F of $\mathbf{B}[\bar{X}]$. $T(\bar{X})$ denotes the set of power products consisting of variables $\bar{X}$.

Definition 8
For an ideal I of a polynomial ring $\mathbf{B}[\bar{X}]$, a finite subset G of I is called a Gröbner basis *of I if $\langle LM(I)\rangle = \langle LM(G)\rangle$.*

Definition 9
For a polynomial $f \in \mathbf{B}[\bar{X}]$, let $a = LC(f)$, $t = LT(f)$ and $h = Rd(f)$. A monomial reduction *$\rightarrow_f$ by f is defined as follows:*

$$bts + p \rightarrow_f (1 - a)bts + absh + p.$$

(Note that $(bts + p) - ((1 - a)bts + absh + p) = bs(af)$.)
Where s is a term of $T(\bar{X})$, b is an element of $\mathbf{B}$ such that $ab \neq 0$ and p is any polynomial of $\mathbf{B}[\bar{X}]$. For a set $F \subseteq \mathbf{B}[\bar{X}]$, we write $g \rightarrow_F g'$ if and only if $g \rightarrow_f g'$ for some $f \in F$. A recursive closure of $\rightarrow_F$ is denoted by $\overset{*}{\rightarrow}_F$, i.e. $g \overset{*}{\rightarrow}_F g'$ if and only if $g = g'$ or there exist a sequence of monomial reductions $g \rightarrow_F g_1 \rightarrow_F \cdots \rightarrow_F g_k = g'$.

Theorem 10
When F is finite, $\rightarrow_F$ is noetherian, that is there is no infinite sequence of polynomials $g_1, g_2, \dots$ such that $g_i \rightarrow_F g_{i+1}$ for each $i = 1, 2, \dots$.

Theorem 11
Let I be an ideal of a polynomial ring $\mathbf{B}[\bar{X}]$.
A finite subset G of I is a Gröbner basis of I if and only if $\forall h \in I \; h \overset{*}{\rightarrow}_G 0$.

Using our monomial reductions, a reduced Gröbner basis is similarly defined as in a polynomial ring over a field. A Gröbner basis G is *reduced* if each polynomial of G is not reducible by a monomial reduction of any other polynomial of G. In a polynomial ring over a field, a reduced Gröbner basis is uniquely determined. In our case, however, this property does not hold.

Example 12
Let $\mathbf{B} = \mathbb{GF}_2 \times \mathbb{GF}_2$. In a polynomial ring $\mathbf{B}[X]$, $\{(1, 0)X, (0, 1)X\}$ and $\{(1, 1)X\}$ are both reduced Gröbner bases of the same ideal.

In order to have a unique Gröbner basis, we need one more definition.

Definition 13
A reduced Gröbner basis G is said to be stratified *if G does not contain two polynomials which have the same leading power product.*

Theorem 14
If G and G' are stratified Gröbner bases of the same ideal w.r.t. some term order, then $G = G'$.

In the above example, $\{(1, 1)X\}$ is the stratified Gröbner basis, but the other is not.

Definition 15
For a polynomial f, $LC(f)f$ is called the Boolean closure *of f, and denoted by $bc(f)$. If $f = bc(f)$, f is said to be* Boolean closed.

Theorem 16
Let G be a Gröbner basis of an ideal I, then $bc(G) \setminus \{0\}$ is also a Gröbner basis of an ideal I.

Theorem 17
Let G be a reduced Gröbner basis, then every element is Boolean closed.

S-polynomial is also defined similarly as in a polynomial ring over a field.

Definition 18
Let $f = atr + f'$ and $g = bsr + g'$ be polynomials where $a = LC(f)$, $b = LC(g)$, $tr = LT(f)$ and $sr = LT(g)$ for some power product t, s, r such that $GCD(t, s) = 1$, i.e. t and s do not contain a common variable. The polynomial $bsf + atg = bsf' + atg'$ is called an S-polynomial *of f and g and denoted by $S(f, g)$.*

As in a polynomial ring over a field, the following property is crucial for the construction of Gröbner bases.

Theorem 19
Let G be a finite set of polynomials such that each element of G is Boolean closed. Then, G is a Gröbner basis if and only if $S(f, g) \overset{*}{\rightarrow}_G 0$ for any pair f, g of G.

For any given finite set F, using our monomial reductions, we can always construct a Gröbner basis of $\langle F\rangle$ by computing Boolean closures and S-polynomials using the following algorithms. It is also easy to construct a stratified Gröbner basis from a Gröbner basis.

Algorithm 20 (BC)
Input: *F a finite subset of $\mathbf{B}[\bar{X}]$*
Output: *F' a set of Boolean closed polynomials such that $\langle F'\rangle = \langle F\rangle$*

```
begin
$F' = \emptyset$
while there exists a polynomial $f \in F$ which is not Boolean closed
      $F = F \cup \{bc(f) - f\} \setminus \{f\}$,  $F' = F' \cup \{bc(f)\}$
end.
```

Algorithm 21 (GBasis)
Input: *F a finite subset of $\mathbf{B}[\bar{X}]$, $>$ a term order of $T(\bar{X})$*
Output: *G a Gröbner basis of $\langle F\rangle$ w.r.t. $>$*

```
begin
$G$ = BC($F$)
while there exists two polynomials $p, q \in G$ such that $S(p,q) \overset{*}{\rightarrow}_G h$
       for some non-zero polynomial $h$ which is irreducible by $\rightarrow_G$
      $G$ = $G \cup$BC($\{h\}$)
end.
```

Since any element of a Boolean ring is idempotent, a Boolean polynomial ring is more natural to work on. We can also define Gröbner bases in Boolean polynomial rings. A power product $X_1^{l_1} \cdots X_n^{l_n}$ is called a *Boolean power product* if each l_i is either 0 or 1. The set of all Boolean power products consisting of variables $\bar{X}$ is denoted by $BT(\bar{X})$. A Boolean polynomial $f(\bar{X})$ in $\mathbf{B}(\bar{X})$ is uniquely represented by $b_1t_1 + \cdots + b_kt_k$ with elements $b_1, \ldots, b_k$ of $\mathbf{B}$ and distinct Boolean power products $t_1, \ldots, t_k$. We call $b_1t_1 + \cdots + b_kt_k$ the *canonical representation* of $f(\bar{X})$. Since $BT(\bar{X})$ is a subset of $T(\bar{X})$, a term order $>$ on $T(\bar{X})$ is also defined on $BT(\bar{X})$. Given a term order $>$, we use the same notations $LT(f), LM(f), LC(f)$ and $Rd(f)$ as before, which are defined by using its canonical representation. We also use the same notations $LT(F)$ and $LM(F)$ for a set F of Boolean polynomials as before.

Definition 22
For an ideal I of a Boolean polynomial ring $\mathbf{B}(\bar{X})$, a finite subset G of I is called a Boolean Gröbner basis *of I if $\langle LM(I)\rangle = \langle LM(G)\rangle$ in $\mathbf{B}(\bar{X})$.*

Using canonical representations of Boolean polynomials, we can also define monomial reductions for Boolean polynomials as Definition 8 and have the same property of Theorem 10. We can also define a stratified Boolean Gröbner basis as in Definition 11, which is unique w.r.t. a term order. The Boolean closure of a Boolean polynomial is also similarly defined as Definition 13 and the same properties of Theorem 14,15 and 17 hold. Construction of a Boolean Gröbner basis is very simple. Given a finite set of Boolean polynomials $F \subseteq \mathbf{B}(\bar{X})$. Compute a Gröbner basis G of the ideal $\langle F \cup \{X_1^2 - X_1, \ldots, X_n^2 - X_n\}\rangle$ in $\mathbf{B}[\bar{X}]$ w.r.t. the same term order. Then, $G \setminus \{X_1^2 - X_1, \ldots, X_n^2 - X_n\}$ is a Boolean Gröbner basis of $\langle F\rangle$ in $\mathbf{B}(\bar{X})$. If G is stratified, then $G \setminus \{X_1^2 - X_1, \ldots, X_n^2 - X_n\}$ is also stratified.
An alternative computation algorithm of Boolean Gröbner bases introduced in [9] is based on the following fact which is essentially a special instance of Theorem 2.3 of [12].

Definition 23
Let $\mathbf{B}$ be a Boolean ring and k be a natural number. $\mathbf{B}^k$ denotes a direct product, i.e. the set of all k-tuples of elements of $\mathbf{B}$. For an element p of $\mathbf{B}^k$, $p_i \in \mathbf{B}$ denotes the i-th element of p for each $i = 1, \ldots, k$. If we define $p + q$ and $p \cdot q$ for $p, q \in \mathbf{B}^k$ by $(p + q)_i = p_i + q_i$ and $(p \cdot q)_i = p_i \cdot q_i$ for each $i = 1, \ldots, k$, $\mathbf{B}^k$ also becomes a Boolean ring. For a polynomial $f(\bar{X})$ in $\mathbf{B}^k[\bar{X}]$ $f_i(i = 1, \ldots, k)$ denotes the polynomial in $\mathbf{B}[\bar{X}]$ obtained by replacing each coefficient p of f by p_i. For a Boolean polynomial $f(\bar{X})$ in $\mathbf{B}^k(\bar{X})$, a Boolean polynomial f_i in $\mathbf{B}(\bar{X})$ is defined similarly.

Theorem 24
In a polynomial ring $\mathbf{B}^k[\bar{X}]$, let G be a finite set of Boolean closed polynomials. Then, G is a (reduced) Gröbner basis of an ideal I if and only if $G_i = \{g_i | g \in G\} \setminus \{0\}$ is a (reduced) Gröbner basis of the ideal $I_i = \{f_i | f \in I\}$ in $\mathbf{B}[\bar{X}]$ for each $i = 1, \ldots, k$.

Corollary 25
In a Boolean polynomial ring $\mathbf{B}^k(\bar{X})$, let G be a finite set of Boolean closed Boolean polynomials. Then, G is a (reduced) Boolean Gröbner basis of an ideal I if and only if $G_i = \{g_i | g \in G\} \setminus \{0\}$ is a (reduced) Gröbner basis of the ideal $I_i = \{f_i | f \in I\}$ in $\mathbf{B}(\bar{X})$ for each $i = 1, \ldots, k$.

Let F be a finite set of polynomials in $\mathbf{B}[\bar{X}]$ for a computable Boolean ring $\mathbf{B}$. Note that a Boolean subring which is generated from the set of all coefficients of polynomials in F is finite. We denote this ring by $\mathbf{B}_F$. By Stone's representation theorem, $\mathbf{B}_F$ is isomorphic to $\mathbb{GF}_2^k$ for some natural number k. Let ψ be such an isomorphism. We also extend ψ to a polynomial ring $\mathbf{B}_F[\bar{X}]$. Identifying a polynomial p of $\mathbf{B}[\bar{X}]$ with its image $\psi(p)$ in $\mathbb{GF}_2^k$, $F_i(i = 1, \ldots, k)$ denotes a set of i-th projection of F in $\mathbb{GF}_2[\bar{X}]$, that is $F_i = \{\psi(p)_i | p \in F\}$. $G_i(i = 1, \ldots, k)$ denotes Gröbner basis of F_i. For each $i = 1, \ldots, k$, let e_i denote an element of $\mathbb{GF}_2^k$ such that its i-th component is 1 and the other component is 0. For a polynomial f in $\mathbb{GF}_2[\bar{X}]$ and each $i = 1, \ldots, k$, $Ex_i(f)$ denotes a polynomial in $\mathbb{GF}_2[\bar{X}]^k$ obtained from f by replacing each monomial t with $e_i t$. Now, the algorithm is given as follows:

Algorithm 26 (AGBasis)
Input: *F a finite subset of $\mathbf{B}[\bar{X}]$, $>$ a term order of $T(\bar{X})$*
Output: *G a Gröbner basis of $\langle F \rangle$ w.r.t. $>$*

```
begin
for each $i = 1, \ldots, k$
$G_i$ = GBasis($F_i, >$)
$G^i$ = $\{Ex_i(g) | g \in G_i\}$
$G$ = $\psi^{-1}(\cup_{i=1}^k G^i)$
end.
```

For a finite set F of Boolean polynomials, its Boolean Gröbner basis is also computed by the same algorithm with a suitable modification.
Note that GBasis$(F_i, >)$ compute a usual Gröbner basis of $\langle F_i \rangle$ in a polynomial ring $\mathbb{GF}_2[\bar{X}]$ over the field $\mathbb{GF}_2$. If each G^i is a reduced Gröbner basis, obviously $\cup_{i=1}^k G_i$ is a reduced Gröbner basis and so is G. It is also easy to construct the stratified Gröbner basis from a reduced Gröbner basis. If we can see k is small(say less than 100) and ψ is easily computable from the input F a priori, this algorithm is more practical than the original one as is reported in [9]. When k is very big, however, this algorithm may be impractical.

4 main result

Let $S = \{s_1, s_2, \ldots, s_k\}$ be a finite set. In this section, we consider a Boolean ring $\mathbf{B}$=$\mathcal{P}(S)$. Our goal is solving a system of equations of a Boolean polynomial ring over $\mathbf{B}(\bar{X})$ with a strong restriction that each variable X_i must be a singleton set. If we do not have such a restriction, we can solve a system of equations simply by computing a Boolean Gröbner basis of the corresponding ideal. A stratified Boolean Gröbner basis G with respect to a purely lexicographic term order is actually a canonical representation of the whole solutions. It has a solution in $\mathbf{B}$ if and only if G does not contain a non-zero constant of $\mathbf{B}$ by Boolean weak Nullstellensatz. In case it has a solution, we can obtain all solutions step by step by solving linear equations with one variable from the smallest variable to the biggest variable, which is an easy consequence of Boolean extension theorem. With the above restriction, however, this method is not sufficient since we cannot describe 'X is a singleton set' as an equation of $\mathbf{B}(\bar{X})$. Thus, we can not solve a system of equations with the restriction by simply computing a Boolean Gröbner basis in general. In order to overcome this difficulty, we show a nice property of Boolean Gröbner bases concerning a certain type of polynomials so called *almost solution polynomials*. Using this result, we can solve a system of equations with the restriction by only computing Boolean Gröbner bases.

Definition 27
Let ϕ be an isomorphism from $\mathcal{P}(S)$ to $(\mathbb{GF}_2)^k$ such that $\phi(\{s_1\}) = (1, 0, \ldots, 0)$, $\phi(\{s_2\})$ =$(0, 1, \ldots, 0)$, …, $\phi(\{s_k\}) = (0, 0, \ldots, 1)$. Let ϕ_j $(j = 1, \ldots, k)$ be a projection homomorphism from $\mathcal{P}(S)$ to $\mathbb{GF}_2$ induced by this isomorphism, that is $\phi_j(T) = 1$ if $s_j \in T$ otherwise $\phi_j(T) = 0$ for each $T \subseteq S$. We also naturally extend ϕ to an isomorphism from $\mathcal{P}(S)(\bar{X})$ to $(\mathbb{GF}_2)^k(\bar{X})$ and ϕ_j to an homomorphism from $\mathcal{P}(S)(\bar{X})$ to $\mathbb{GF}_2(\bar{X})$.

Definition 28
For any variable X_i and element s_j, a Boolean polynomial $f(\bar{X})$ is called an almost solution polynomials *of X_i with respect to s_j if it satisfies the following condition* (i). *A Boolean polynomial $g(\bar{X})$ is also called* an almost solution polynomials *of X_i with respect to s_j if it satisfies the following condition* (ii).

(i) $\phi_j(f(\bar{X})) = X_i + 1$,

(ii) $\phi_t(g(\bar{X})) = X_i$ *for every t except for j.*

For an almost solution polynomial of X_i w.r.t. s_j, $X_i + \{s_j\}$ is called its associated solution polynomial.

Lemma 29
Let I be an ideal of a Boolean polynomial ring $\mathcal{P}(S)(\bar{X})$ and I does not contain constants. If I contain an almost solution polynomial of X_i w.r.t. s_j, then the value of X_i of any singleton solution must be $\{s_j\}$.

proof Since $\phi_j(\{s_j\})$, $\phi_j(f(\bar{X})) = \phi_j(\{s_j\}f(\bar{X}))$.
If $\phi_j(\{s_j\}f(\bar{X})) = X_i+1$, $\{s_j\}f(\bar{X})$ must be equal to $\{s_j\}X_i+\{s_j\}$ since a coefficient of every monomial of $\{s_j\}f(\bar{X})$ is equal to $\{s_j\}$. Since $\{s_j\}f(\bar{X})$ is also contained in I, $\{s_j\}X_i + \{s_j\} \in I$. $\{s_j\}X_i + \{s_j\} = 0$ is equivalent to $\{s_j\} \subseteq X_i$. By a similar reason, if $\phi_t(f(\bar{X})) = X_i$ for every t except for j, $\{s_t\}X_i = 0$ which is equivalent to $\{s_t\} \cap X_i = \emptyset$, i.e. $\{s_t\} \notin X_i$. □

We will show that if an ideal contains almost solution polynomials we can easily obtain all of them from an arbitrary stratified Boolean Gröbner basis. We first observe the following easy fact.

Theorem 30
Let I be an ideal of a Boolean polynomial ring $\mathbb{GF}_2(\bar{X})$ and I is not a trivial ideal $\langle I\rangle$. Let G be a reduced Boolean Gröbner bases of I with respect to any term order. If I contains a polynomial consisting of one variable, then G contains this polynomial.

proof Suppose a non-trivial ideal I contains $X + a$(a is 0 or 1). Then $X + a \overset{*}{\rightarrow}_G 0$. Therefore G must contain a Boolean polynomial $X + h(\bar{X})$ such that $LT(X + h(\bar{X})) = X$. Since $h(\bar{X}) - a \in I$, $h(\bar{X}) \overset{*}{\rightarrow}_G a$. $h(\bar{X})$ must be equal to a because G is reduced. □

Now, we are ready to prove our main result.

Theorem 31
Let I be an ideal of a Boolean polynomial ring $\mathcal{P}(S)(\bar{X})$ and I does not contain constants. Let G be a stratified Boolean Gröbner basis of I with respect to any term order. For any variable X_i and

element s_j, if I contains an almost solution polynomial of X_i w.r.t. s_j, then G contains an almost solution polynomial of X_i w.r.t. s_j.

proof By Theorem 20, $\phi_j(G) = \{\phi_j(g) | g \in G\}$ is the reduced Gröbner basis of $\phi_j(I) = \{\phi_j(f) | f \in I\}$. Since, $\mathcal{P}(S)$ is isomorphic to $(\mathbb{GF}_2)^k$, the assertion follows from Theorem 25. (Take $\mathbf{B}$=$\mathbb{GF}_2$ in Theorem 20.) □

Example 32
The left constraint with unknown set variables X and Y is equivalent to the right system of equations of a Boolean polynomial ring $\mathcal{P}(S)(X, Y)$ for $S = \{s_1, s_2, \ldots, s_k\}$

$$\begin{cases} X \cup Y \subseteq \{s_1, s_2\} \\ s_1 \in X \\ X \cap Y = \emptyset \end{cases} \iff \begin{cases} (1 + \{s_1, s_2\})(XY + X + Y) = 0 \\ \{s_1\}X + \{s_1\} = 0 \\ XY = 0 \end{cases}$$

Let $F = \{(1 + \{s_1, s_2\})(XY + X + Y), \{s_1\}X + \{s_1\}, XY\}$. A Boolean Gröbner basis G of $\langle F\rangle$ with respect to the purely lexicographic term order of $Y > X$ is $\{\{s_2\}XY, (1 + \{s_2\})Y, (1 + \{s_2\})X + \{s_1\}\}$. $(1 + \{s_2\})Y$ is an almost solution polynomial of Y w.r.t. s_2 with its associated solution polynomial $Y + \{s_2\}$. $(1 + \{s_2\})X + \{s_1\}\}$ is an almost solution polynomial of X w.r.t. s_1 with its associated solution polynomial $X + \{s_1\}$. Adding these associated solution polynomials to F, we have its stratified Boolean Gröbner basis $\{X + \{s_1\}, Y + \{s_2\}\}$.

We may not always find almost solution polynomials in Boolean Gröbner bases.

Example 33
Consider the followng set constraint.

$$\begin{cases} X \cup Y \subseteq \{s_1, s_2, s_3\} \\ s_1 \in X \\ X \cap Y = \emptyset \end{cases} \iff \begin{cases} (1 + \{s_1, s_2, s_3\})(XY + X + Y) = 0 \\ \{s_1\}X + \{s_1\} = 0 \\ XY = 0 \end{cases}$$

A Boolean Gröbner basis G of $\langle F\rangle$ for $F = \{1+\{s_1, s_2, s_3\})(XY+X+Y), \{s_1\}X+\{s_1\}, XY\}$ with respect to the purely lexicographic term order of $Y > X$ is $\{\{s_2, s_3\}XY, (1 + \{s_2, s_3\})Y, (1 + \{s_2, s_3\})X + \{s_1\}\}$. $(1+\{s_2, s_3\})X+\{s_1\}\}$ is the only almost solution polynomial in G. Its associated solution polynomial is $X + \{s_1\}$. Adding it to F, we have its stratified Boolean Gröbner basis $\{X + \{s_1\}, 1 + \{s_2, s_3\}Y\}$. It does not contain almost solution polynomials of Y.

Even when we encounter a situation as in the last example, we can still go further by adding all possible singleton solutions. In the above examples, possible singleton solutions of Y is either $\{s_2\}$ or $\{s_3\}$, add $Y + \{s_2\}$ or $Y + \{s_3\}$ to F and compute the stratified Gröbner basis, if it does not contain constant it is a solution. In this case both of them are solutions. Now we are ready to present an algorithm to solve singleton set constraints with only computations of Boolean Gröbner bases.

Algorithm 34 (`SingSolve`)
Input: *F a finite subset of $\mathcal{P}(\{s_1, s_2, \ldots, s_k\})(\bar{X})$*
Output: *E the set of all singleton solutions of the system of equations $\{f = 0 \mid f \in F\}$*

```
begin
$E = \{\}$
$G$ = SingSolveMain($F$)
if $G$ contains only solution polynomials, then $E = E \cup \{G\}$,
else
   if $G$ contains constants, then terminate,
   else pick a variable $X$ such that $G$ does not contain a solution
         polynomial of $X$, let $X + \{s_{p_1}\}, \ldots, X + \{s_{p_l}\}$ be all the
```

```
        possible solution polynomials of $X$ and
        $E = E \cup$SingSolve($G \cup \{X + \{s_{p_i}\}\}$) for each $i = 1, \ldots, l$
end.
```

Algorithm 35 (SingSolveMain)
Input: *F a finite subset of $\mathcal{P}(\{s_1, s_2, \ldots, s_k\})(\bar{X})$*
Output: *G a finite subset of $\mathcal{P}(\{s_1, s_2, \ldots, s_k\})(\bar{X})$*

```
begin
$G$ = GBasis($F$,$>$) for any term order $>$ of $T(\bar{X})$
if $G$ contains constants, then terminate,
else
   if $G$ contains almost solution polynomials,
      then replace each almost solution polynomial with its
      solution polynomial in $G$ and $G$ =SingSolveMain($G$),
end.
```

There is also a naive approach for solving a singleton set constraint by Boolean Gröbner bases. Once we have obtained a stratified Boolean Gröbner basis of the corresponding ideal w.r.t. a purely lexicographic term order, using it we can get whole solutions by solving linear equations of one variable step by step from the lowest variable to the highest variable. According to Theorem 26, if the ideal contains an almost solution polynomial, any Boolean Gröbner basis also contains this information. However, in the above naive method, we do not use this information except for the information of the lowest variable at each step. Therefore our algorithm SingSolve should be more efficient than the naive method. In order to check this, we have also made a program for the naive method. Comparison between the above algorithm and the naive method will be shown in the next section.

5 Application for Sudoku puzzles

We have developed a software to compute Boolean Gröbner bases [1] in the computer algebra system Risa/Asir [3]. The software also contains a program for our new algorithm. In this section, we show that our algorithm is satisfactorily practical for for solving Sudoku puzzles. Sudoku is one of the most popular puzzle in the world. This puzzle can be considered as a singleton set constraint.

4				9				
3						1	8	
			5					
		5	8					
	2	9						
					1	7		
		6					5	
					7			
			2					9

Consider the above sudoku puzzle. We associate a variable X_{ij} for each grid at the i-th row and the j-th column. This puzzle can be considered as a set constraint where each variable should be assigned a singleton set from 9 candidates $\{1\}, \{2\}, \ldots, \{9\}$ so that any distinct two variables which lie on a same row, column or block must be assigned different singleton sets. 17 variables are assigned singleton sets $X_{11} = \{4\}, X_{15} = \{9\}, \ldots, X_{99} = \{9\}$ as the initial conditions. This constraint is translated into a system of equations of a Boolean polynomial ring $\mathbf{B}(X_{11}, X_{12}, \ldots, X_{99})$ with $\mathbf{B} = \mathcal{P}(\{1, 2, \ldots, 9\})$ as follows:

(1) $X_{11} = \{4\}, X_{15} = \{9\}, \ldots, X_{99} = \{9\}$.
(2) $X_{ij}X_{i'j'} = 0(= \emptyset)$ for each pair of distinct variables $X_{ij}, X_{i'j'}$ which lie on a same row, column or block.
(3) $\sum_{(i,j)\in A} X_{ij} = 1(= \{1, 2, \ldots, 9\})$ where A is a set of indices lying on a same row, column or block. (There are 27 such A's.)

This puzzle is nothing but solving the above equations with a strong restriction that each variable must be a singleton set.

We have solved many(more than 1000) Sudoku puzzles by our implementation of `SingSolve`. All of them are solved within practical time. We show some timing data in the following list. The column 'SingSolve' contains data by `SingSolve`, the column 'Naive' contains data by our program of the naive method described at the end of the last section. We can use any term order in `SingSolve`, however, we used a purly lexichographic term order $X_{99} > \cdots > X_{91} > \cdots > X_{11}$ for both computations in order to make the comparison fair. In each computation `SingSolve` of data 1, the algorithm does not call the last procedure `elso`, whereas each computation `SingSolve` of data 2 calls the last procedure `elso` at least once. All sudoku puzzles have 17 initial conditions. Our machine's detail is OS Windows Vista and CPU Intel Core2Duo 3.06GHz with SDRAM 8GB. The running time is measured in seconds.

puzzle	Naive	SingSolve
1	366.78	18.11
2	489.21	19.19
3	903.84	20.17
4	677.01	19.58
5	542.88	19.47
6	1578.84	19.75
7	923.51	16.47
8	530.12	15.66
9	1501,99	16.27
10	2383.33	22.37

computation data 1

puzzle	Naive	SingSolve
11	5040.51	15.97
12	2632.54	19.34
13	5543.07	20.55
14	4262	24.34
15	2005.96	18.67
16	6677.45	26.02
17	784.97	27.94
18	2312.2	18.24
19	3273.04	22.06
20	1878.37	17.58

computation data 2

6 Conclusion

Our purpose is not to make a fast program for a typical combinatorial problem but to make a general program which can handle many types of combinatorial problems. Actually there exist much faster programs for Sudoku puzzles. However, there does not exists a single program which can handle 9×9 Sudoku puzzles and 16×16 Sudoku puzzles. Our program can handle not only both of them, but also any variant of Sudoku puzzles such as diagonal Sudoku.

There does not exist a mathematical formulation concerning the ranks of difficulty of Sudoku puzzles. Our approach could give such mathematical formulation. Most Sudoku puzzles which do not need the last `else` procedure in `SingSolve` are not ranked difficult in the source of the puzzles, on the other hand, most puzzles ranked as the highest difficulty need the last `else` procedure.

References and Notes

[1] Inoue, S.(2009). BGSet - a software to compute Boolean Gröbner bases -. http://www.mi.kagu.tus.ac.jp/ inoue/BGSet

[2] Inoue, S.(2009). On the Computation of Comprehensive Boolean Gröbner Bases. Proceedings of the 11th International Workshop on Computer Algebra in Scientific Computing(CASC 2009), LNCS 5743, pp 130-141, Springer-Verlag Berlin Heidelberg.

[3] Noro, M. et al. (2009). A Computer Algebra System Risa/Asir.
http://www.math.kobe-u.ac.jp/Asir/asir.html.

[4] Rudeanu, S. Boolean functions and equations. North-Holland Publishing Co.,Amsterdam-London; American Elsevier Publishing Co.,Inc.,New York,1974

[5] Sakai, K. and Sato, Y. (1988). Boolean Gröbner bases. ICOT Technical Momorandum 488.
http://www.icot.or.jp/ARCHIVE/Museum/TRTM/tm-list-E.html

[6] Sakai, K., Sato, Y. and Menju, S. (1991). Boolean Gröbner bases(revised). ICOT Technical Report 613.
http://www.jipdec.or.jp/icot/ARCHIVE/Museum/TRTM/tr0613.htm

[7] Sato, Y. et al.(1995). Set Constrains Solvers(Prolog version).
http://www.jipdec.or.jp/icot/ARCHIVE/Museum/FUNDING/funding-95-E.html

[8] Sato, Y.(1998). A new type of canonical Gröbner bases in polynomial rings over Von Neumann regular rings. Proceedings of ISSAC 1998, ACM Press, pp 317-32.

[9] Sato, Y. et al.(1998). Set Constrains Solvers(Klic version).
http://www.jipdec.or.jp/icot/ARCHIVE/Museum/FUNDING/funding-98-E.html

[10] Sato, Y. and Inoue, S.(2005). On the Construction of Comprehensive Boolean Gröbner Bases. Proceedings of the 7th Asian Symposium on Computer Mathematics(ASCM 2005), pp 145-148.

[11] Sato, Y., Nagai, A. and Inoue, I.(2008). On the Computation of Elimination Ideals of Boolean Polynomial Rings, Proceedings of the 8th Asian Symposium on Computer Mathematics(ASCM 2007), LNAI 5081, pp 334-348, Springer-Verlag Berlin Heidelberg.

[12] Weispfenning, V. (1989). Gröbner bases in polynomial ideals over commutative regular rings. In Davenport Ed., editor, *EUROCAL'87*, pp 336-347. Springer LNCS 378, 1989.