A REDUCTION OF 3-SAT PROBLEM TO BUCHBERGER ALGORITHM.

MAIIA BAKHOVA

Abstract. There is a number of known NP class problems, and majority of them have been shown to be equivalent to others. In particular now it is clear that constructing of a Gröbner basis must be one of equivalent problems, but there was no example. In the following paper the reduction is constructed.

1. Introduction

3-SAT problem. Let $x_1, x_2, \ldots, x_k$ denote boolean variables (their value may be True or False). A literal is either a variable or its negation. An equation in the propositional calculus is an expression that can be constructed using literals and the operations and, denoted by $\land$, and or, denoted by $\lor$. An example of such equations is $(x_1 \land \neg x_2) \lor (\neg x_3 \land x_4) = \text{False}$. We will use exponents 1 and $-1$ to indicate if a variable is with negation or not; so $x_i^1 := x_i$ and $x_i^{-1} := \neg x_i$. Any Boolean equation has an equivalent form as

$$\bigwedge_{i=1}^{n} (x_{i1}^{\sigma_{i1}} \lor x_{i2}^{\sigma_{i2}} \lor \ldots \lor x_{im}^{\sigma_{im}}) = \text{True}.$$ 

The problem of finding solution for such equation is called the Boolean satisfiability problem. It has been shown that any satisfiability problem can be transformed in polynomial number of steps (adding variables as needed) into an equation

$$\bigwedge_{i=1}^{n} (x_{i1}^{\sigma_{i1}} \lor x_{i2}^{\sigma_{i2}} \lor x_{i3}^{\sigma_{i3}}) = \text{True}.$$ 

The corresponding problem is called the 3-SAT problem. There are some obvious cases in which we can see immediately if a solution exists or not. For example if the number of clauses is fewer than 8 some solution always exists. From the other hand, if we consider a maximal

1991 Mathematics Subject Classification. 14Q20, 68Q15,13P10.
number of clauses for a one solution, say when all variables are true, then the number of clauses which are true is

\[
\frac{2k(2k-2)(2k-4)}{6} - \frac{k(k-1)(k-2)}{6} = \frac{7k(k-1)(k-2)}{6},
\]

hence if there are more clauses, then a solution does not exists.

3-SAT problem is one of so called NP problems, see [1]. There are different versions of required answer for such problem. One of them is to find a solution which satisfies the Boolean equation. Another is Yes/No, meaning that one should indicate if any solution exists.

Gröbner basis. Gröbner basis is defined as a basis of ideal in multivariate polynomial ring generated by a given set of polynomials, see [2]. The basis is required to have specific properties which help quickly and efficiently to resolve a number of questions about the ideal. For example, it allows to get quick answers about its zero locus. In particular, if the zero locus is empty set, then the corresponding Gröbner basis equals 1. The process of finding a Gröbner basis is called Buchberger algorithm.

2. The Reduction

Consider a usual 3-SAT problem, which is a Boolean equation with \(k\) variables \(x_1, x_2, \ldots, x_k\). Then the 3-SAT problem can be transformed into a system of \(n\) Boolean equations:

\[
\begin{align*}
&x_{11}^{\sigma_{11}} \lor x_{12}^{\sigma_{12}} \lor x_{13}^{\sigma_{13}} = \text{True} \\
&\cdots \\
&x_{n_1}^{\sigma_{n_1}} \lor x_{n_2}^{\sigma_{n_2}} \lor x_{n_3}^{\sigma_{n_3}} = \text{True}
\end{align*}
\]

All variables \(x_{i_1}, \ldots, x_{i_3}\) (usually with repetitions) in the system are from the set \(\{x_1, x_2, \ldots, x_k\}\). All exponents can admit only values 1 or -1.

Conversion procedure. We will replace the system of Boolean equations with a system of real polynomial equations with the following formal process. Introduce new real valued variables \(z_1, z_2, \ldots, z_k\). I want each appearance of a literal \(x_{ij}^{\sigma_{ij}}\) replace with \((z_{ij} - 1)\) or \(z_{ij}\) depending on the value of exponent \(\sigma_{ij}\). For this purpose we introduce constants \(c_{i_1}, \ldots, c_{i_3}\), which values are defined by exponents:

\[
c_{ij} = \begin{cases} 
1, & \text{if } \sigma_{ij} = 1 \\
0, & \text{if } \sigma_{ij} = -1
\end{cases}
\]

Now let us substitute \(x_{ij}^{\sigma_{ij}}\) with \((z_{ij} - c_{ij})\). Replace a boolean operation \(\lor\) with a usual multiplication and the "True" value with 0. Now the
system looks like a polynomial equation system:

\[
\begin{align*}
(z_{i1} - c_{i1})(z_{i2} - c_{i2})(z_{i3} - c_{i3}) &= 0 \\
\vdots \\
(z_{n1} - c_{n1})(z_{i2} - c_{i2})(z_{i3} - c_{i3}) &= 0
\end{align*}
\]

We say that such system is associated to the system of Boolean equations \(2.1\).

**Lemma 2.1.** A set of Boolean values \(\{R_1, R_2, \ldots, R_k\}\) is a solution for 3-Sat problem iff the number set \(\{r_1, r_2, \ldots, r_k\}\) is a solution for the associated polynomial equation system, where \(r_i = 1\) if \(R_i = \text{True}\) and \(r_i = 0\) if \(R_i = \text{False}\) for all \(i = 1, 2, \ldots, k\).

**Proof.** Without loss of generality we can assume that the problem \(2.1\) includes an equation \(x_{11} \lor x_{22} \lor x_{33} = \text{True}\). The associated polynomial equation will be \((z_1 - c_1)(z_2 - c_2)(z_3 - c_3) = 0\).

Let us consider a case when \(\sigma_1 = 1\). Observe that when we assign a Boolean variable \(x_{11}\) to be True, and the corresponding variable \(z_1\) is 1, then the clause on the right is True, and the polynomial at the left side of the equation associated to such clause vanishes, because it has a factor \((z_1 - 1)\). In the case when the Boolean variable is False, then the value of the clause is defined by values of other variables. The corresponding factor in polynomial expression is non zero and the whole expression can vanish only if there is other factor which is zero. The same is true in reverse when we start with assigned values for variables of the polynomial system. When the numerical variable \(z_1\) is 1, then the associated polynomial vanishes. The corresponding Boolean variable \(x_{11} = \text{True}\) and the initial Boolean equation is true.

The case \(\sigma_1 = -1\) is treated similarly. Now we can generalize the reasoning for all clause equations. \(\square\)

**Corollary 2.2.** A 3-Sat problem has a solution iff the corresponding polynomial equation system has a numeric solution.

**Proof.** This is trivial. \(\square\)

**Theorem 2.3.** A 3-SAT problem can be resolved using Buchberger algorithm.

**Proof.** Given a 3-SAT problem construct the associated system of polynomial equations. Take all polynomials on the left hand side of the equations and use them to define an ideal. Now use Buchberger algorithm to find solutions. The question about the existence of the solution will be answered, and there are different approaches to find a particular solution. \(\square\)
Note that the same reduction can be done for a 2-SAT problem and it yields a quadratic equation system. Then choosing a specific value of one of the variable turns some equations into linear ones with simple solution. The process forces other variables to take specific values or show that they can take any value. This is similar to how the equivalent graph problem is solved. Now the reason why the 3-SAT and 2-SAT problems are so different becomes apparent in polynomial interpretation, which answers a question of Pr. E. M. Luks raised in a private discussion.

ACKNOWLEDGMENTS

The author thanks E. M. Luks for introduction to the problem and helpful discussions.

REFERENCES

[1] Michael R. Garey and David S. Johnson, in Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, 1979.

[2] Martin Kreuzer, Lorenzo Robbiano, in Computational Commutative Algebra 1, Springer, Oxford University Press, 2008.

MATHEMATICS DEPARTMENT, BATON ROUGE COMMUNITY COLLEGE, BATON ROUGE, LOUISIANA

E-mail address: majabakh@gmail.com