AN ALGEBRAIC APPROACH TO THE BOOLEAN SATISFIABILITY PROBLEM

ZACHARY ROMRELL

1. Motivation

When learning about logic in CSCI 373 this past semester I started to sense a connection between the process of computing a grobner basis and an algorithm lectured for determining whether a logical sentence in conjunctive normal form is satisfiable or not. With more thought I realized that one can quite easily rephrase any propositional logic sentence as a polynomial living in $\mathbb{F}_2[x_1, \ldots, x_n]$. With this rephrasing by considering the vanishing of the set of polynomials corresponding to the Boolean satisfiability problems of interest one can use algebraic techniques to determine if there exists a model for the satisfiability problem (i.e a solution $(a_1, \ldots, a_n), a_i \in \mathbb{F}_2$). When searching the literature for such a connection I found a few discussion posts regarding this relationship and a master's thesis from the University of Groningen [2] further studying this connection citing Ideals, Varieties, and Algorithms by Cox et. at. and using a majority of theorems and definitions introduced throughout our course. In Section 2, I will first give a brief overview of the boolean satisfiability problem. Later in Section 3, I will describe how we can represent satisfiability problems algebraically using some results described in [2]. Lastly in Section 4, we will walk through some examples of relating instances of the satisfiability problem to algebraic geometry and discuss more relationships to the theory being lectured in CSCI 373.

2. Introduction to the Boolean Satisfiability Problem

In logic and computer science, the Boolean satisfiability problem is a problem of determining if there exists an interpretation that satisfies a given Boolean formula. Essentially it asks whether a set of Boolean variables describing a Boolean formula can be set to true and false in a way such that the overall formula evaluates to true.

Consider the countable set of Boolean variables $\Sigma = \{\sigma_i | i \in \mathbb{N}\}$. We can define a propositional language $L(\Sigma)$ as the smallest set of strings such that:

1. $\top \in L(\Sigma)$ and $\bot \in L(\Sigma)$ (where $\top$ represents an always true formula and $\bot$ represents an always false formula)
2. $\sigma_i \in \Sigma$, then $\sigma_i \in L(\Sigma)$
3. $\sigma_i \in L(\Sigma)$, then $\neg\sigma_i \in L(\Sigma)$
4. if $\sigma_j, \sigma_k \in L(\Sigma)$, then $(\sigma_j \vee \sigma_k) \in L(\Sigma)$
5. if $\sigma_j, \sigma_k \in L(\Sigma)$, then $(\sigma_j \wedge \sigma_k) \in L(\Sigma)$

where $\neg$, $\vee$, and $\wedge$ are defined as follows

$$\neg\sigma_i = \begin{cases} true & if\, \sigma_i = false \\ false & otherwise \end{cases} \tag{2.0.1}$$

$$(\sigma_j \vee \sigma_k) = \begin{cases} true & if\, \sigma_j = true\ or\ \sigma_k = true \\ false & otherwise \end{cases} \tag{2.0.2}$$

$$(\sigma_j \wedge \sigma_k) = \begin{cases} true & if\, \sigma_j = true\ and\ \sigma_k = true \\ false & otherwise \end{cases} \tag{2.0.3}$$

it is also important to note that there exists many properties with how the logical connectives ($\neg, \vee, \wedge$) defined above relate strings in $L(\Sigma)$. Namely commutativity $a \vee b = b \vee a$, associativity $(a \vee b) \vee c = a \vee (b \vee c)$, distributivity $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$, De Morgan's laws ($\neg(a \vee b) = (\neg a \vee \neg b)$), and other properties that can be proven using the basic tools already introduced.

We can also relate sentences in a propositional language to a semantic space, denoted $S(\Sigma)$ which is a space defining all possible sets of models (true/false assignments). For example if $\Sigma = \{a, b\}$ the possible models in the model space, $M(\Sigma)$, would be $a = 1, b = 1$ denoted ab, $a = 1, b = 0$ denoted $a\overline{b}$, $a = 0, b = 1$ denoted $\overline{a}b$, $a = 0, b = 0$ denoted $\overline{a}\overline{b}$ and the elements in the semantic space would be the power set of $\{ab, a\overline{b}, \overline{a}b, \overline{a}\overline{b}\}$, $\mathcal{P}(\{ab, a\overline{b}, \overline{a}b, \overline{a}\overline{b}\})$. Therefore, $|M(\Sigma)| = 2^{|\Sigma|}$ and $|S(\Sigma)| = 2^{|M(\Sigma)|}$

A given propositional sentence can satisfy any number of these models. We will denote the mapping from the propositional language to the semantic space with $I : L(\Sigma) \mapsto S(\Sigma)$. The interpretation, I, of a string of the propositional language $L(\Sigma)$ is defined:

(1) $I(\top) = M(\Sigma)$
(2) $I(\bot) = \emptyset$
(3) if $\sigma_i \in \Sigma$, then $I(\sigma_i) = \{m | m \in M(\Sigma), m(\sigma_i) = 1\}$
(4) if $\sigma_i \in \Sigma$, then $I(\neg\sigma_i) = \overline{I(\sigma_i)}$
(5) if $\sigma_j, \sigma_k \in L(\Sigma)$, then $I((\sigma_j \vee \sigma_k)) = I(\sigma_j) \cup I(\sigma_k)$
(6) if $\sigma_j, \sigma_k \in L(\Sigma)$, then $I((\sigma_j \wedge \sigma_k)) = I(\sigma_j) \cap I(\sigma_k)$

For example when $\Sigma = \{a, b\}, I(\top) = \{ab, a\overline{b}, \overline{a}b, \overline{a}\overline{b}\} = M(\Sigma), I(a) = \{ab, a\overline{b}\}, I(\neg a) = \{\overline{a}b, \overline{a}\overline{b}\}, I(a \wedge \neg b) = \{a\overline{b}\}$, etc. This notation above was not introduced in [2], however, will be useful in Section 4 when further analyzing this relationship and to better understand the algebraic equivalence of resolution inference and Gödel's completeness theorem.

3. Relationship to Algebraic Geometry

I will now discuss how we can interpret Boolean satisfiability problems algebraically using some results from [2]. Throughout the majority of the semester when working in the world of ideals and varieties we were considering polynomials in $k[x_1, \ldots, x_n]$ where k was an infinite field. However, for the algebraic variant of satisfiability problems we will work with polynomials in $\mathbb{F}_2[x_1, \ldots, x_n]$. In this polynomial ring we can still use much of the theory of ideals and varieties in the same way as we have when working with infinite fields. In this setting $x_1, \ldots, x_n$ would correspond to the Boolean variables $\sigma_i \in \Sigma$. The assignment of $\sigma_i = true$ corresponds to $x_i = 0$ and $\sigma_i = false$ corresponds to $x_i = 1$. In terms of translating a propositional sentence into a polynomial in $\mathbb{F}_2[x_1, \ldots, x_n]$ we will do this as follows ([2] does this in a different way, however, this approach seems more straightforward and intuitive).

$$\sigma_i = x_i \tag{3.0.1}$$

$$\neg\sigma_i = x_i + 1 \tag{3.0.2}$$

$$(\sigma_j \vee \sigma_k) = (x_j * x_k) \tag{3.0.3}$$

$$(\sigma_j \wedge \sigma_k) = (x_j + x_k) \tag{3.0.4}$$

For example consider the propositional sentence $(a \wedge b) \vee \neg c$ where $\Sigma = \{a, b, c\}$. From the mapping above the corresponding polynomial would be $(x + y) * (z + 1) \in \mathbb{F}_2[x, y, z]$. As you can see when setting c to false (i.e $z = 1$) the following result will be a multiple of 2 and therefore vanish and indicate a solution to the Boolean expression. This leads to the following definition:

Definition 3.1. $f \in \mathbb{F}_2$ is *satisfiable* if there exists a point $(a_1, \ldots, a_n) \in \mathbb{F}_2^n$ such that $f(a_1, \ldots, a_n) = 0$.

However, since only a single solution is required to make a propositional sentence satisfiable, we can also view this problem in the realm of varieties. Namely, when $V(f) \neq \emptyset$ we would want f to be satisfiable. This observation suggests that we can utilize ideals along with an analog of the Weak Nullstellensatz in $\mathbb{F}_2$ to potentially find an algorithmic way of determining whether such a satisfiability problem has a solution. Unfortunately, the weak Nullstellansatz presented in our textbook requires k to be an algebraically closed field, however, by having a predetermined set of generators in the ideal along with the polynomial analog of the Boolean satisfiability problem one can prove a special case for the weak Nullstellansatz of $\mathbb{F}_2$. First recall the following definition of a radical ideal.

Definition 3.2. The *radical* of an ideal I in $k[x_1, \ldots, x_n]$, denoted by $\sqrt{I}$, is defined as:

$$\sqrt{I} = \{f \in k[x_1, \ldots, x_n] | f^n \in I \text{ for some integer } n \geq 1\}$$

An ideal I is *radical* if $\sqrt{I} = I$. Using the definitions above one can prove the following lemma.

Lemma 3.3. *$I \subset R$ is radical if and only if R/I does not contain nonzero nilpotents.*

Using Lemma 3.3 and the third isomorphism theorem, Theorem 3.4 follows and any ideal $I = \langle f, x_1^2 + x_1, \ldots, x_n^2 + x_n \rangle$ is radical.

Theorem 3.4. *For each ideal $I \subseteq \mathbb{F}_2[x_1, \ldots, x_n] = R$ such that $x_j^2 + x_j \in I$ for each i, R/I does not contain nonzero nilpotents.*

The above results from [2] also correspond nicely to a weaker analog of the weak Nullstellansatz over finite field described in [1].

Theorem 3.5. *Let $f \in \mathbb{F}_2[x_1, \ldots, x_n]$ and let Γ_2 denote the ideal of $\mathbb{F}_2[x_1, \ldots, x_n]$ generated by $\langle x_1^2 + x_1, \ldots, x_n^2 + x_n \rangle$ then:*

(1) f vanishes at all of $\mathbb{F}_2^n \iff f \in \Gamma_2$

(2) Let $f_1, \ldots, f_m \in \mathbb{F}_2[x_1, \ldots, x_n]$. If $f \in \boldsymbol{I}(\boldsymbol{V}(f_1, \ldots, f_m))$ then $f = g_1 f_1 + \ldots + g_m f_m + \gamma$ for some $\gamma \in \Gamma_2$

Therefore, by considering the ideal $I = \langle f, x_1^2 + x_1, \ldots, x_n^2 + x_n \rangle$ for our algebraic equivalent boolean expression f, the weak Nullstellansatz cannot immediately be applied since $\mathbb{F}_2$ is not algebraically closed, however, we can introduce its closure and prove a special case given ideals of the form above.

$$\overline{\mathbb{F}_2} = \bigcup_{n=1}^{\infty} \mathbb{F}_{2^{n!}}$$

By now considering the ideal $\overline{I} = \langle f, x_1^2 + x_1, \ldots, x_n^2 + x_n \rangle$ in $\overline{\mathbb{F}_2}[x_1, \ldots, x_n]$. It follows from the weak Nullstellansatz that $\mathbf{V}(\overline{I}) = \emptyset \iff 1 \in \overline{I}$. This observation leads to the main result from [2] for making this connection between Boolean satisfiability problems and its algebraic representation.

Theorem 3.6. *Let $\overline{\mathbb{F}_2}$ be the algebraic closure of $\mathbb{F}_2$. Given $f \in \mathbb{F}_2[x_1, \ldots, x_n]$ consider the ideals $I = \langle f, x_1^2 + x_1, \ldots, x_n^2 + x_n \rangle$ in $\mathbb{F}_2[x_1, \ldots, x_n]$ and $\overline{I} = \langle f, x_1^2 + x_1, \ldots, x_n^2 + x_n \rangle$ in $\overline{\mathbb{F}_2}[x_1, \ldots, x_n]$. Then $\mathbf{V}(I) = \emptyset$ if and only if $\mathbf{V}(\overline{I}) = \emptyset$.*

An immediate corollary to this theorem is as follows.

Corollary 3.7. *$\boldsymbol{V}(I) = \emptyset$ if and only if $1 \in I$.*

Therefore, we can now redefine the condition that will allow us to utilize Buchberger's algorithm to determine whether a propositional logic sentence is indeed satisfiable.

Definition 3.8. The polynomial $f \in \mathbb{F}_2[x_1, \ldots, x_n]$ is unsatisfiable if and only if $1 \in I = \langle f, x_1^2 + x_1, \ldots, x_n^2 + x_n \rangle$.

In propositional logic there exists two popular sentence forms, DNFs and CNFs. By using properties of logic every propositional formula can be put in CNF form which can also be transformed to DNF form and vice versa. However, in the polynomial realm what does this imply? Perhaps a polynomial representing a CNF and a polynomial representing a logically equivalent DNF may have a different structure. It was proven in [2] that when transforming these polynomials to $\mathbb{F}_2[x_1, \dots, x_n]/(x_1^2 + x_1, \dots, x_n^2 + x_n)$ they are equivalent. In other words for equivalent formulas P_{CNF} and P_{DNF}, the bases $\{\overline{f}_{P_{CNF}}, x_1^2 + x_1, \dots, x_n^2 + x_n\}$ and $\{\overline{f}_{P_{DNF}}, x_1^2 + x_1, \dots, x_n^2 + x_n\}$ are the same.

Finally, we can ask if every polynomial $f \in \mathbb{F}_2[x_1, \dots, x_n]/(x_1^2 + x_1, \dots, x_n^2 + x_n)$ represents a propositional logic formula. By providing a translation from polynomial operators to propositional logic we can find the propositional logic formula corresponding to an operation on f, g in $\mathbb{F}_2[x_1, \dots, x_n]/(x_1^2 + x_1, \dots, x_n^2 + x_n)$

$$P_{f+g} = P_f \wedge P_g \tag{3.0.5}$$

$$P_{f \cdot g} = P_f \vee P_g \tag{3.0.6}$$

Therefore, every polynomial $f \in \mathbb{F}_2[x_1, \dots, x_n]/(x_1^2 + x_1, \dots, x_n^2 + x_n)$ corresponds to a propositional logic formula. [2] concludes with two algorithms for solving SAT involving computing Gröbner. In Section 4 we will further relate the previously introduced logic terminology to its algebraic equivalent form.

4. Further Relationships

We will first touch on the idea of logical inference. Given $\Sigma = \{a, b, c\}$ can we infer $b \vee c$ from $(a \vee b \vee c) \wedge \neg a$? Yes, because $I((a \vee b \vee c) \wedge \neg a) = \{\overline{a}\overline{b}c, \overline{a}b\overline{c}, \overline{a}bc\}$. While $I(b \vee c) = \{\overline{a}\overline{b}c, \overline{a}b\overline{c}, \overline{a}bc, a\overline{b}c, ab\overline{c}, abc\}$ and $\{\overline{a}\overline{b}c, \overline{a}b\overline{c}, \overline{a}bc\} \subseteq \{\overline{a}\overline{b}c, \overline{a}b\overline{c}, \overline{a}bc, a\overline{b}c, ab\overline{c}, abc\}$. More generally for any $\alpha, \beta \in L(\Sigma), \alpha \vDash \beta$ if and only if $I(\alpha) \subseteq I(\beta)$. We also have that a sentence $\alpha \in L(\Sigma)$ is satisfiable if its interpretation has at least one model (i.e $I(\alpha) \neq \emptyset$). However, what does all this mean in the algebraic world?

The first immediate connect when working in $\mathbb{F}_2[x_1, \dots, x_n]$ is that $|\Sigma| = n$. Therefore, there is a one-to-one correspondence between the number of variables in your polynomial ring and the number of literals in your logical signature. It also immediately follows that $|\mathbb{F}_2^n| = |M(\Sigma)|$, namely every point in the affine space of $\mathbb{F}_2^n$ corresponds to an assignment of true/false values for the Boolean literals. In logic we also have this notion of an interpretation function, I, which maps $L(\Sigma)$ to $S(\Sigma)$ where the semantic space is the power set of $M(\Sigma)$. This is equivalent to taking the vanishing of the algebraic equivalent propositional sentence. This is because an affine variety is the set of all solutions satisfying its defining polynomials and the number of unique varieties will be all possible subsets of the affine space (2^{k^n}). In the case of $\mathbb{F}_2$ this would be 2^{2^n}. The semantic space on the other hand is exactly this, $|S(\Sigma)| = 2^{|M(\Sigma)|}$ where $|M(\Sigma)| = 2^{|\Sigma|}$, therefore, $|S(\Sigma)| = 2^{2^{|\Sigma|}}$. For example, when $\Sigma = \{a, b\}$ then $\neg a$ corresponds to the following polynomial $x + 1$ in $\mathbb{F}_2[x, y]$. In logic $I(\neg a) = \{\overline{a}b, \overline{a}\overline{b}\}$. Similarly we can consider $\mathbf{V}(x + 1) = \{(1, 0), (1, 1)\} \subseteq \mathbb{F}_2^2$ (recall setting a to false corresponds to setting $x = 1$). Now that we have established this equivalence what is the algebraic analog of the inference question asked before?

Recall $\alpha \vDash \beta$ (alpha entails beta) if and only if $I(\alpha) \subseteq I(\beta)$. Letting the polynomials corresponding to α, β be P_α, P_β, algebraically this implies $\mathbf{V}(P_\alpha) \subseteq \mathbf{V}(P_\beta)$. In other words $P_\alpha \in \mathbf{V}(P_\beta)$. Therefore, in the realm of ideals $\mathbf{I}(\mathbf{V}(P_\alpha)) \supseteq \mathbf{I}(\mathbf{V}(P_\beta))$. In logic there is a resolution inference algorithm for determining whether a satisfiability problem entails false. Essentially with this algebraic analog the algorithm breaks up a CNF sentences (subset of propositional sentences) by its clauses (split at each $\wedge$), therefore, algebraically this would correspond to a polynomial being split into monomials (by its addition signs). The algorithm then finds propositional sentences that are entailed by the monomials of the original polynomial. In terms of varieties, the algorithm is finding the smallest variety containing the varieties of the two propositional sentences you can perform resolution on. At any step if two logical clauses imply the empty variety, we immediately know the original propositional sentence is unsatisfiable. From Theorem 3.6 this would also imply the ideal corresponding to this polynomial and $x_1^2 + x_1, \ldots, x_n^2 + x_n$ is $\langle 1 \rangle$. This algorithm relies on the proof of Gödel's completeness theorem which would be interesting to algebraically study.

5. Conclusion

In conclusion, the theory we have learned throughout the year is super relevant to framing and analyzing real world problems. The following resolution algorithm also has substantially different run times depending on the number of clauses in the propositional sentence and the number of corresponding literals. This relies on the number of models a specific propositional sentence is probabilistically likely to realize given these parameters. Therefore, the run time would be the worse for maximal ideals in $\mathbb{F}_2[x_1, \ldots, x_n]$. Overall, analyzing this connection between algebraic geometry and Boolean satisfiability problems has been interesting and I am confident there are a lot of cool results yet to be found regarding this relationship.

References

[1] Sudhir R. Ghorpade. A note on nullstellensatz over finite fields, 2018.

[2] B. Klooster. An algebraic approach to the boolean satisfiability problem. *Master Project Mathematics*, 2016.

Department of Mathematics, Williams College,Williamstown, MA 01267
Email address: zr3@williams.edu