VERIFICATION OF SEQUENTIAL GALOIS FIELD CIRCUITS USING ALGEBRAIC GEOMETRY

Xiaojun Sun*, Priyank Kalla*
*Electrical & Computer Engineering, University of Utah

I. FURTHER DISCUSSION ON COMPUTER ALGEBRA

Bit level circuit variables including primary inputs/outputs can only take value within $\mathbb{Z}_2$. They can also be written in functions of other variables and constants. In computer algebra, circuit variable a can be described as a polynomial, that is $a = f(x_0, x_1, \ldots, x_{n-1}) \in \mathbb{F}[x]$. Moreover, circuit variable can have multiple specifications, they can be written in polynomials and included in an ideal $I = \{f_0, f_1, \ldots, f_k\}$. The varieties of this ideal are possible assignments of circuit variables. Similarly for n bits world-level circuit variables, the varieties are elements from $\mathbb{F}_{2^k}$. Thus it's convenient to use ideals and varieties to describle circuit variables.

Definition 1.1: (**Sum of Ideals**) If I and J are ideals in $k[x_1, \ldots, x_n]$, then the **sum** of I and J, denoted by $I+J$, is the set

$$I + J = \{f + g : f \in I \text{ and } g \in J\}. \tag{1}$$

Furthermore, if $I = \langle f_1, \ldots, f_r \rangle$ and $J = \langle g_1, \ldots, g_s \rangle$, then $I + J = \langle f_1, \ldots, f_r, g_1, \ldots, g_s \rangle$.

Definition 1.2: (**Product of Ideals**) If I and J are ideals in $k[x_1, \ldots, x_n]$, then the **product** of I and J, denoted by $I \cdot J$, is defined to be the ideal generated by all polynomials $f \cdot g$ where $f \in I$ and $g \in J$. Furthermore, let $I = \langle f_1, \ldots, f_r \rangle$ and $J = \langle g_1, \ldots, g_s \rangle$, then

$$I \cdot J = \langle f_i g_j : 1 \le i \le r, 1 \le j \le s \rangle. \tag{2}$$

Definition 1.3: (**Quotient of Ideals**) If I and J are ideals in $k[x_1, \ldots, x_n]$, then $I : J$ is the set

$$\{f \in k[x_1, \ldots, x_n] : fg \in I, \forall g \in J\} \tag{3}$$

and is called the **ideal quotient** of I by J.

According to definitions above, it is possible to calculate ideal sum, product and quotient based on ideal generators. Additional conclusion about union, intersection and complement of varieties can be attained below:

Theorem 1.1: If I and J are ideals in $k[x_1, \ldots, x_n]$, then $\mathbf{V}(I + J) = \mathbf{V}(I) \bigcap \mathbf{V}(J)$ and $\mathbf{V}(I \cdot J) = \mathbf{V}(I) \bigcup \mathbf{V}(J)$.

Theorem 1.2: Let V and W be varieties in $\mathbb{F}_{2^k}$. Then $\mathbf{I}(V) : \mathbf{I}(W) = \mathbf{I}(V - W)$.

For 1.2, let $V = \mathbb{F}_{2^k} = \mathbf{V}(\langle \textit{vanishing polynomials} \rangle)$, then $V - W$ is the **complementary set** of variety W.

Based on these conclusions it's easy to get a circuit variable's specifications or possible assignments from the other known information.

II. SEQUENTIAL CIRCUITS AND FINITE STATE MACHINE

A. Introduction to Finite State Machine:

Borrow from somewhere.

B. Symbolic Image Computation:

Borrow from somewhere.

C. Implicit State Enumeration:

BFS algorithm below:

ALGORITHM 1: Breadth-first Traversal Algorithm

Input: Transition functions Δ, initial state S^0
$from^0 = reached = S^0$;
repeat
  $i \leftarrow i + 1$;
  $to^i \leftarrow \text{Img}(\Delta, from^{i-1})$;
  $new^i \leftarrow to^i \cdot reached$;
  $reached \leftarrow reached + new^i$;
  $from^i \leftarrow new^i$;
until $new^i == 0$;
return *reached*

III. EXPERIMENTS USING NEW APPROACH

First example is reachable state enumeration of a 2-bit finite state machine. Breadth-First-Search traversal (mentioned in II) is adopted.

Example 3.1: Sample circuit is described below with gate-level implement and state transition graph.

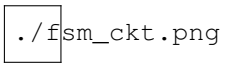

Fig. 1: Sample circuit

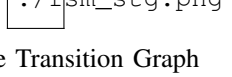

Fig. 2: State Transition Graph

The following mapping from Boolean operations to Galois field $\mathbb{F}_2$ operations is straightforward. Assuming a and b are both bit-level variables:

- $\overline{a} \rightarrow 1 + a$
- $a \text{ and } b \rightarrow ab$
- $a \text{ or } b \rightarrow a + b + ab$

Comparing to the quantification approach in II, using Gröbner bases approach from **??**. Construct an elimination ideal with following polynomials: $f_1 : 1 + XNOR(t_0, OR(p \cdot q, (1 + x_i) \cdot (1 + p) \cdot (1 + q)))$; $f_2 : 1 + XNOR(t1, OR(xi \cdot (1 + p), p \cdot (1 + q)))$; $f_3 : S - p - q \cdot \alpha$; $f_4 : T - t_0 - t_1 \cdot \alpha$ and vanishing polynomials $f_5 : p^2 - p$; $f_6 : q^2 - q$; $f_7 : t_0^2 - t_0$; $f_8 : t_1^2 - t_1$; $f_9 : S^4 - S$; $f_{10} : x_i^2 - x_i$; $f_{11} : T^4 - T$, where S and T are 2-bit word representing current state and next state as defined in f_3 and f_4, f_1 and

f_2 describe the transition relations. At last the initial state should also be included in this elimination ideal. For example initial state is S_3, then last polynomial is $f_{12} : S + 1 + \alpha$. According to elimination theorem **??**, let the lex ordering be $\{all\ others\} > T$, calculate Gröbner basis, the result must contain a polynomial with only one variable T, which can indicate the possible assignments of next state.

Apply this approach to calculate image in BFS traversal **??**. In this example, to^i and *reached* are both ideals containing only one generator, i.e. the polynomial with single variable T; universal set is varieties from ideal $\langle T^4 - T \rangle$. Use 1.1 and 1.2 to complete **??**, the final return value is polynomial $to^3 = T^4 + T$, which means all states are rechable from state S_3.

The second example shows the application of the new approach on sequential arithematic circuits function verification.

Example 3.2:

The following design is Sequential Multiplier with Parallel Output (SMPO), a Normal Basis multiplier based on Massey-Omura algorithm. Inputs and outputs are all 5-bit word taking value from $\mathbb{F}_{2^5}$. After loading operands A and B, setting all output latches to 0 and running for 6 iterations, the output should be $R = A \cdot B \pmod{x^5 + x^2 + 1}$.

Fig. 3: 5-bit SMPO

Similarly, build elimination ideal for all gates/operations and induce initial states of latches. However, instead of eliminating all variables to one, this example adopts abstraction term order from **??** and keeps the polynomial contains function between output and inputs. Here the lex ordering is $\{others\} > R > \{A, B\}$, and objective polynimial is $R + \mathcal{F}(A, B)$.

Apply this approach to calculate image in BFS traversal, but modify **??** to make it adapt 6 steps reachable states enumeration. The result is $R + AB$, which validates the function of this circuit.