Part I

Algebraic Structures

1

Compositions and Clones of Boolean Functions

Reinhard Pöschel and Ivo Rosenberg

1.1 Boolean Polynomials

The representations of Boolean functions are frequently based on the fundamental operations $\{\vee, \wedge, '\}$, where the disjunction $x \vee y$ represents the logical OR, the conjunction $x \wedge y$ represents the logical AND and is often denoted by $x \cdot y$ or simply by the juxtaposition xy, and x' stands for the negation, or complement, of x and is often denoted by $\overline{x}$. This system naturally appeals to logicians and, for some reasons, also to electrical engineers, as illustrated by many chapters of this volume and by the monograph [7]. Its popularity may be explained by the validity of many identities or laws: for example, the associativity, commutativity, idempotence, distributive, and De Morgan laws making $\mathcal{B} := \langle B; \vee, \wedge, ', 0, 1\rangle$ a Boolean algebra, where $B = \{0, 1\}$; in fact, $\mathcal{B}$ is the least nontrivial Boolean algebra.

It is natural to ask whether there is a system of basic Boolean functions other than $\{\vee, \wedge, '\}$, but equally powerful in the sense that each Boolean function may be represented over this system. To get such a system, we introduce the following binary (i.e., two-variable) Boolean function $\dot{+}$ defined by setting $x \dot{+} y = 0$ if $x = y$ and $x \dot{+} y = 1$ if $x \neq y$; its truth table is

x	y	$x \dot{+} y$
0	0	0
0	1	1
1	0	1
1	1	0

Clearly $x \dot{+} y = 1$ if and only if the arithmetical sum $x + y$ is odd, and for this reason $\dot{+}$ is also referred to as the sum *mod* 2. It corresponds to the "exclusive or" of logic, whereby the "exclusive or" of two statements P and Q is true if "either P or Q" is true. Notice that some natural languages, such as, French, distinguish "or" from "exclusive or" ("ou" et "soit"), whereas most natural languages are less

precise. The function $\dot{+}$ is also denoted $\oplus$ or $+$ and, in more recent engineering literature, by EXOR.

Consider the system $\{\dot{+}, \cdot, 0, 1\}$, where 0 and 1 are constants. A reader familiar with groups will notice that $\langle B; \dot{+}\rangle$ is an abelian group with neutral element 0 satisfying $x\dot{+}x \approx 0$ (also called an elementary 2-group), where $\approx$ stands for an identity on B. Moreover, $GF(2) := \langle B; \dot{+}, \cdot, 0, 1\rangle$ is a field, called a Galois field and denoted $\mathbf{Z}_2$ or F_2. Thus, in $GF(2)$ we may use all the arithmetic properties valid in familiar fields (such as the fields $\mathbf{Q}$, $\mathbf{R}$, and $\mathbf{C}$ of all rational, real, and complex numbers) but not their order or topological properties. In addition, $GF(2)$ also satisfies $x\dot{+}x \approx 0$ and $x^2 \approx x$. Clearly $GF(2)$ is the field of the least possible size, and so one may be inclined to dismiss it as a trivial and unimportant object. Surprisingly, it has serious applications. A practical one is in cryptography and coding theory (for secret or secure data transmission, for example, for governments, banks, or from satellites; see Chapters 8 and 9 in this volume).

Denote $x_1\dot{+}\cdots\dot{+}x_n$ by $\dot{\sum}_{i=1}^{n} x_i$. Let f be an n-ary Boolean function distinct from the constant c_0^n (which is the n-variable Boolean function with constant value 0). In its complete disjunctive normal form (DNF), replace the disjunction $\vee$ (of complete elementary conjunctions) by their sum mod 2 $\dot{\sum}$. This is still a representation of f because for every $(a_1, \ldots, a_n) \in \mathbf{B}^n$, at most one of the elementary conjunctions takes value 1 and $1\dot{+}0\dot{+}\cdots\dot{+}0 = 1$ and $0\dot{+}0\dot{+}\cdots\dot{+}0 = 0$. Using $x^0 \approx x' \approx 1\dot{+}x$ and $x^1 \approx x$ throughout, we obtain a representation of f over $\{\dot{+}, \cdot, 0, 1\}$. The following proposition makes this more precise. Here the symbol $\prod$ denotes the usual arithmetical product. We make the usual convention that in expressions involving $\dot{+}, \cdot$ and 1 products are calculated before sums, for example, $xy\dot{+}z$ stands for $(xy)\dot{+}z$, and that $\prod$ and $\dot{\sum}$ over the empty set are 1 and 0, respectively.

Proposition 1.1. [28] *For every n-ary Boolean function f, there exists a unique family F of subsets of $N = \{1, \ldots, n\}$ such that*

$$f(x_1, \ldots, x_n) \approx \dot{\sum_{I\in F}} \prod_{i\in I} x_i. \tag{1.1}$$

For example, $x_1 \wedge x_2 \approx x_1\dot{+}x_2\dot{+}x_1x_2$ with $F = \{\{1\}, \{2\}, \{1, 2\}\}$ (direct verification). Call the right-hand side of (1.1) a Boolean polynomial.

Proof. Let f be an n-ary Boolean function. If $f = c_0^n$, take $F = \emptyset$. Thus, let $f \neq c_0^n$. In the discussion leading to the proposition, we saw that f may be represented over $\{\dot{+}, \cdot, 0, 1\}$. Multiplying out the parentheses, we obtain a representation of f as a polynomial in variables $x_1, \ldots, x_n$ over $GF(2)$. In view of $x^2 \approx x$, it may be reduced to a sum of square-free monomials. From $x\dot{+}x \approx 0$, it follows that it may be further reduced to such a sum in which every monomial appears at most once, proving the representability of f by a Boolean polynomial. It remains to prove the uniqueness. For every $F \subseteq \mathcal{P}(N)$, that is, a family of subsets of N, denote

by $\varphi(F)$ the corresponding Boolean polynomial. Clearly φ is a map from the set $\mathcal{P}(\mathcal{P}(N))$ of families of subsets of N into the set $O^{(n)}$ of n-ary Boolean functions.

Claim. *The map φ is injective.*

Indeed, by the way of contradiction, suppose $\varphi(F) = \varphi(G)$ for some $F, G \subseteq \mathcal{P}(N)$ with $F \neq G$. Choose $I \in (F \setminus G) \cup (G \setminus F)$ of the least possible cardinality, say, $I \in F \setminus G$. Put $a_i = 1$ for $i \in I$ and $a_i = 0$ otherwise. Then for $\underline{a} = (a_1, \ldots, a_n)$, it is easy to see that $\varphi(F)(\underline{a}) = \varphi(G)(\underline{a}) \dot{+} 1$ (as every subset of I is either in both families F and G or in neither). This contradiction shows $G = F$. Now $|\mathcal{P}(\mathcal{P}(N))| = 2^{2^n} = |O^{(n)}|$, and hence φ is a bijection from $\mathcal{P}(\mathcal{P}(N))$ onto $O^{(n)}$, proving the uniqueness. ■

Remark 1.1. *The representation from Proposition 1.1 is sometimes referred to as the Reed-Muller expression or the algebraic normal form of f (see, e.g., Chapter 8). So far, Boolean polynomials have been less frequently used than the disjunctive and conjunctive normal forms, but they have proved indispensable in certain theoretical studies, such as enumeration or coding theory. More recently, electrical engineers have also become interested in Boolean polynomials.*

Remark 1.2. *Boolean polynomials may be manipulated in a conceptually simple way. For example, using the representations $x_1 \vee x_2 \approx x_1 \dot{+} x_2 \dot{+} x_1 x_2$ and $x_1 \to x_2 \approx 1 \dot{+} x_1 \dot{+} x_1 x_2$, we can compute*

$$\begin{aligned}(x_1 \wedge x_2)(x_1 \to x_2) &= (x_1 \dot{+} x_2 \dot{+} x_1 x_2)(1 \dot{+} x_1 \dot{+} x_1 x_2) \\ &= x_1 \dot{+} x_2 \dot{+} x_1 x_2 \dot{+} x_1^2 \dot{+} x_1 x_2 \dot{+} x_1^2 x_2 \dot{+} x_1^2 x_2 \dot{+} x_1 x_2^2 \dot{+} x_1^2 x_2^2 \\ &= x_1 \dot{+} x_2 \dot{+} x_1 x_2 \dot{+} x_1 \dot{+} x_1 x_2 \dot{+} x_1 x_2 \dot{+} x_1 x_2 \dot{+} x_1 x_2 \dot{+} x_1 x_2 \\ &= x_2.\end{aligned}$$

This can be easily performed by a computer program, but we may face an explosion in the number of terms.

Remark 1.3. *Suppose that an n-ary Boolean function is given by a table. How do we find its Boolean polynomial or, equivalently, the corresponding family F? We could proceed via the complete DNF (as indicated earlier), but this again may produce a large number of monomials at the intermediate stages. A direct algorithm is as follows.*

Let $f \in O^{(n)}$. On the hypercube B^n we have the standard (partial) order: $(a_1, \ldots, a_n) \leq (b_1, \ldots, b_n)$ if $a_1 \leq b_1, \ldots, a_n \leq b_n$; for example; for $n = 2$ we have that $(1, 0) \leq (1, 1)$, but $(1, 0) \leq (0, 1)$ does not hold. As usual, we write $\underline{a} < \underline{b}$ if $\underline{a} \leq \underline{b}$ and $\underline{a} \neq \underline{b}$. For $i = 1, \ldots, n$, set $S_i = \{(a_1, \ldots, a_n) \in B^n : a_1 + \cdots + a_n = i\}$; hence, S_i consists of all $\underline{a} \in B^n$ having exactly i coordinates equal to 1. Recursively, we construct $H_i \subseteq S_i$ $(i = 0, \ldots, n)$. Set $H_0 = S_0$ if $f(0, \ldots, 0) = 1$, and $H_0 = \emptyset$ otherwise. Suppose $0 \leq i < n$ and $H_0, \ldots, H_i$ have been constructed. For $\underline{a} \in S_{i+1}$, set

$$T_{\underline{a}} = \{\underline{b} \in H_0 \cup \cdots \cup H_i \ : \underline{b} \leq \underline{a}\}$$

and

$$H_{i+1} = \{\underline{a} \in S_{i+1} : f(\underline{a}) + |T_{\underline{a}}| \text{ is odd}\}.$$

For $(a_1, \dots, a_n) \in B^n$, *set*

$$\chi(\underline{a}) = \{1 \le i \le n : a_i = 1\}$$

and set $F = \chi(H_0 \cup \dots \cup H_n)$. *A straightforward proof shows that*

$$f(\underline{x}) \approx \dot{\sum_{I \in F}} \prod_{i \in I} x_i.$$

For example, if $n = 2$ *and* f *is the implication* $\to$, *we get* $H_0 = \{(0,0)\}$, $H_1 = \{(1,0)\}$, $H_2 = \{(1,1)\}$, $F = \{\emptyset, \{1\}, \{1,2\}\}$, *and* $x_1 \to x_2 \approx 1 \dot{+} x_1 \dot{+} x_1 x_2$.

Remark 1.4. *Contrary to disjunctive normal forms, there is no minimization problem for representations of the form (1.1). However, the Boolean polynomials of some Boolean functions are long. The worst one is the Boolean polynomial of* $f(x_1, \dots, x_n) \approx x_1' \dots x_n'$, *whose family* F *is the whole* $\mathcal{P}(N)$; *for example, for* $n = 2$ *we have* $x_1'x_2' \approx 1 \dot{+} x_1 \dot{+} x_2 \dot{+} x_1x_2$. *If, along with* $\dot{+}, \cdot, 1$, *we also allow the negation, certain Boolean polynomials may be shortened. Here we can use some additional rules such as* $x'y \approx y \dot{+} xy$ *and* $xx' \approx 0$. *For these more general polynomials, we face minimization problems. It seems that these and the systematic use of the more general polynomials have not been investigated in depth.*

Remark 1.5. *As the reader may suspect, Proposition 1.1 can be extended to any finite field* $\mathbf{F}$ *(e.g.,* $\mathbf{Z}_3$*) and maps* $f : \mathbf{F}^n \to \mathbf{F}$.

Remark 1.6. *A long list of practical problems (mostly from operations research) leads to the following problem. Let* f *be a map from* B^n *into the set* $\mathbf{Z}$ *of integers and assume that we want to find the minimum value of* f *on* B^n. *The potential of Boolean polynomials for this problem was realized quite early in [6]. A more direct variant and a related duality are in [23, 24], but generally this approach seems to be dormant.*

1.2 Completeness and Maximal Clones

It is well known that every Boolean function may be represented over $\{\wedge, \cdot, '\}$ (e.g., through a DNF or a conjunctive normal form, CNF). In Section 1.1, we saw a representation of Boolean functions over $\{\dot{+}, \cdot, 1\}$ (through Boolean polynomials). In general, call a set F of Boolean functions *complete* if every Boolean function is a composition of functions from F. In some older and East European literature, a complete set is often referred to as *functionally complete*. In universal algebra, the corresponding algebra $\langle B; F\rangle$ is termed *primal*.

Naturally we may ask about other complete sets and their size. The two examples we have seen so far consist of three functions each. However, the set $\{\wedge, \cdot, '\}$ is

actually redundant. Indeed, $\{\wedge, \ '\}$ is also complete as $x_1x_2 \approx (x_1' \wedge x_2')'$ (this identity is known as one of the De Morgan laws).

A Boolean function f is *Sheffer* if the singleton set $\{f\}$ is complete. Clearly a Sheffer function is at least binary. The following functions NAND and NOR are Sheffer:

$$x \text{ NAND } y :\approx x' \vee y', \quad x \text{ NOR } y :\approx x' \wedge y'.$$

The suggestive symbols NAND and NOR were adopted by electrical engineers to describe the functioning of certain transistor gates after these were invented in the 1950s. However, the two functions were introduced in logic long ago by Sheffer [25] and Nicod [19]; they are known as Sheffer strokes or Nicod connectives and are variously denoted by $|, \perp, \top, \uparrow, \downarrow$, and so forth. The fact that NAND is Sheffer follows from

$$x' \approx x \text{ NAND } x, \quad x \wedge y \approx x' \text{ NAND } y';$$

consequently, the complete set $\{\wedge, \ '\}$ can be constructed from NAND alone, and hence NAND is Sheffer. The proof for NOR is similar. The fact that NAND and NOR are the only binary Sheffer functions will follow from Corollary 1.5, which provides a complete characterization of Sheffer functions.

Having seen a few complete sets of Boolean functions, we may ask about other complete sets or – even better – for a completeness criterion, that is, for a necessary and sufficient condition for a set F of Boolean functions to be complete. E. L. Post found such a criterion in [20]. To formulate the criterion in a modern way, we need two crucial concepts. For $1 \leq i \leq n$, the ith *n-ary projection* is the Boolean n-ary function e_i^n satisfying $e_i^n(x_1, \ldots, x_n) \approx x_i$. Notice that e_i^n just replicates its ith argument and ignores all other arguments. The *composition* of an m-ary Boolean function f with the n-ary Boolean functions $g_1, \ldots, g_m$ is the n-ary Boolean function h, defined by setting $h(x_1, \ldots, x_n) = f(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$ for all n-tuples $(x_1, \ldots, x_n)$ in B^n.

Now, a set of Boolean functions is a clone if it is composition-closed and contains all projections. Thus, a clone is rich enough in the sense that we cannot exit from it via any composition of its members (each possibly used several times). The concept is similar to the concept of a transformation monoid (whereby the set of all projections plays a role analogous to that of the identity selfmap in monoids).

It is not difficult to check that the intersection of an arbitrary set of clones is again a clone. Thus, for every set F of Boolean functions, there exists a unique least clone containing F. This clone, denoted by $\langle F \rangle$, is the clone generated by F.

The clone $\langle F \rangle$ can be alternatively interpreted in terms of combinatorial switching circuits. Suppose that for each n-ary $f \in F$ there is a gate (switching device, typically a transistor) with n inputs and a single output realizing $f(a_1, \ldots, a_n)$ on the output whenever, for $i = 1, \ldots, n$, the zero-one input a_i is applied to the ith input. An *F-based combinatorial circuit* is obtained from gates realizing functions from F by attaching the output of each gate (with a single exception) to an input of another gate so that (1) the circuit has a single external output; (2) to each input of

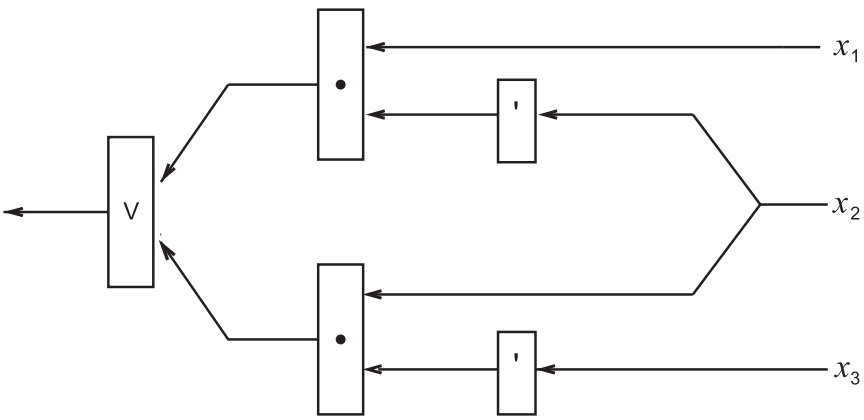

Figure 1.1. Combinatorial circuit realizing $f(x_1, x_2, x_3) \approx x_1 x_2' \vee x_2 x_3'$.

a gate is attached a single external input or a single output of another gate but not both; and (3) the circuit is feedback-free (i.e., following the arcs from any external input, we always arrive at the external output without ever making a loop). An example is in Figure 1.1; see also Chapters 11 and 16.

Now the clone $\langle F \rangle$ generated by F consists of all Boolean functions realizable by F-based combinatorial circuits. This interpretation also allows us to present an alternative definition of the concept of completeness: in this terminology, a set F of Boolean functions is complete exactly if $\langle F \rangle$ is the set O of all Boolean functions. Clearly O is the greatest clone (with respect to inclusion $\subseteq$). A clone C distinct from O is maximal (also precomplete or preprimal) if $C \subset D \subset O$ holds for no clone D: in other words, if O covers C in the containment relation $\subseteq$. Thus, C is maximal exactly if C is incomplete, but $C \cup \{f\}$ is complete for each $f \in O \setminus C$. The concept is a direct analog of a maximal subgroup, a maximal subring, and so on.

We prove in Theorem 1.2 that each clone distinct from O is contained in at least one maximal clone. This leads to the following almost immediate but still basic fact.

Fact 1. *A set F of Boolean functions is complete if and only if F is contained in no maximal clone.*

Proof. ($\Rightarrow$) By contradiction, if F is contained in some maximal clone M, then the clone $\langle F \rangle$ generated by F is included in M, and so F is not complete.
($\Leftarrow$) If F is incomplete, then $\langle F \rangle$ extends to a maximal clone M, and so $F \subseteq \langle F \rangle \subseteq M$. ■

Of course, Fact 1 is fully applicable only if we know the entire list of maximal clones. Then a set F of Boolean functions is complete if and only if for each maximal clone M (from the list) we can find some $f \in F \setminus M$.

Most clones and all maximal clones may be described by relations in the following way. Recall that for a positive integer h, an *h-ary relation* on B is a subset ρ of B^h (i.e., a set of zero-one h-tuples; some logicians prefer to view it as a map φ from B^h into $\{+, -\}$, whereby $\rho = \varphi^{-1}(+)$, or to call it a predicate). For $h = 1$, the relation is called *unary* and is just a subset of B, whereas for $h = 2$, the relation is *binary*.

An n-ary Boolean function f preserves an h-ary relation ρ if for every $h \times n$ matrix A whose column vectors all belong to ρ, the h-tuple of the values of f on the rows of A belongs to ρ. In symbols, if $A = [a_{ij}]$ with $(a_{1j}, \dots, a_{hj}) \in \rho$ for all $j = 1, \dots, n$, then

$$(f(a_{11}, \dots, a_{1n}), \dots, f(a_{h1}, \dots, a_{hn})) \in \rho.$$

Notice the "rows versus columns" type of the definition. Several examples are given next. In the universal algebra terminology, f preserves ρ if ρ is a subuniverse of $\langle B; f\rangle^h$. The first impression may be that the definition is too artificial or covers only special cases. It turns out that it is the right one not only for clones of Boolean functions but also for clones on any finite universe. It seems that it was first explicitly formulated in [15], and it has been reinvented under various names. We illustrate this crucial concept using a few examples needed for the promised completeness criterion.

Example 1.1. *Let $h = 1$ and $\rho = \{0\}$. Then $A = [0 \cdots 0]$ (a $1 \times n$ matrix), and f preserves $\{0\}$ if and only if $f(0, \dots, 0) = 0$ (in universal algebra terminology, if and only if $\{0\}$ is a subuniverse of $\langle B; f\rangle$).*

Example 1.2. *Similarly, a Boolean function f preserves the unary relation $\{1\}$ if and only if $f(1, \dots, 1) = 1$.*

Example 1.3. *Consider the binary relation $\sigma := \{(0, 1), (1, 0)\}$ on B. Note that σ is the diagram (or graph) π^0 of the permutation $\pi : x \mapsto x'$. It can also be viewed as a graph on B with the single edge $\{0, 1\}$. A $2 \times n$ matrix $A = [a_{ij}]$ has all columns in σ if and only if $a_{2j} = a'_{1j}$ holds for all $j = 1, \dots, n$. Writing a_j for a_{1j}, we obtain that f preserves σ if and only if*

$$(f(a_1, \dots, a_n), f(a'_1, \dots, a'_n)) \in \sigma$$

holds for all $a_1, \dots, a_n \in B$. This is equivalent to the identity

$$f(x'_1, \dots, x'_n) \approx f(x_1, \dots, x_n)'.$$

The standard name for a Boolean function satisfying this identity is selfdual *(see [7] and Section 1.3). Expressed algebraically, f is selfdual if and only if the negation $'$ is an automorphism of the algebra $\langle B; f\rangle$. A selfdual $f \in O^{(n)}$ is fully determined by its values $f(0, a_2, \dots, a_n)$ with $a_2, \dots, a_n \in B$ (because $f(1, b'_2, \dots, b'_n) = f'(0, b'_2, \dots, b'_n)$). Thus, there are exactly $2^{2^{n-1}}$ selfdual n-ary Boolean functions, and the probability that a randomly chosen $f \in O^{(n)}$ is selfdual is the low value $2^{-2^{n-1}}$: for example, it is approximately 0.0000152 for $n = 5$.*

Example 1.4. *Next consider* $\rho = \{(0,0),(0,1),(1,1)\}$. *Notice that* $(x,y) \in \rho$ *if and only if* $x \le y$ *(where* $\le$ *is the natural order on* B*), and so we write* $x \le y$ *instead of* $(x,y) \in \rho$. *A* $2 \times n$ *matrix* $A = [a_{ij}]$ *has all columns in* $\le$ *whenever* $a_{1j} \le a_{2j}$ *holds for all* $j = 1, \dots, n$, *and therefore* f *preserves* $\le$ *if and only if*

$$a_1 \le b_1, \dots, a_n \le b_n \Rightarrow f(a_1, \dots, a_n) \le f(b_1, \dots, b_n);$$

that is, if every argument is kept the same or increased, then the value is the same or increases. This is the standard definition of a monotone (also isotone, order-respecting, or order-compatible) Boolean function; see [7].

Example 1.5. *As a final example, consider the 4-ary (or quaternary) relation*

$$\lambda = \{(x_1, x_2, x_3, x_1 \dot{+} x_2 \dot{+} x_3) : x_1, x_2, x_3 \in B\},$$

where $\dot{+}$ *is the sum mod 2 introduced in Section 1.1. Expressed differently, the last coordinate in a 4-tuple from* λ *is exactly the parity check (making the coordinate sum even), a basic error check used in computer hardware and other digital devices; equivalently, a 4-tuple belongs to* λ *if and only if it contains an even number of 1s. The description of the functions preserving* λ *is not as transparent as in the preceding examples. In order to establish it, let us say that a Boolean function* $f \in O^{(n)}$ *is linear (or affine) if there are* $c \in B$ *and* $1 \le i_1 < \ldots < i_k \le n$ *such that* $f(x_1, \dots, x_n) \approx c \dot{+} x_{i_1} \dot{+} \cdots \dot{+} x_{i_k}$.

Fact 2. *A Boolean function preserves* λ *if and only if it is linear.*

Proof. ($\Rightarrow$) Let $f \in O^{(n)}$ preserve λ. Then

$$f(x_1, \dots, x_n) \approx f(x_1, 0, \dots, 0) \dot{+} f(0, x_2, \dots, x_n) \dot{+} f(0, \dots, 0, 0). \tag{1.2}$$

Here $f(0, x_2, \dots, x_n) \in O^{(n-1)}$ also preserves λ, and so we can apply (1.2) to it. Continuing in this fashion, we obtain

$$f(x_1, \dots, x_n) \approx \dot{\sum_{i=1}^{n}} f(0, \dots, 0, x_i, 0, \dots, 0) \dot{+} d \tag{1.3}$$

where $d = f(0, \dots, 0) \dot{+} \ldots \dot{+} f(0, \dots, 0)$ (n times). Each unary Boolean function $f(0, \dots, 0, x_i, 0, \dots, 0)$ is of the form $a_i x \dot{+} b_i$ for some $a_i, b_i \in B$, and thus the right-hand side of (1.3) is $a_1 x_1 \dot{+} \cdots \dot{+} a_n x_n \dot{+} c$ where $c = b_1 \dot{+} \cdots \dot{+} b_n \dot{+} d$. This proves that f is linear.
($\Leftarrow$) It can be easily verified that every linear function preserves λ. ∎

The set of Boolean functions preserving a given h-ary relation ρ on B is denoted by Pol ρ. It is easy to verify that Pol ρ is a clone by showing that it contains all projections and that it is composition closed.

Now we are ready for the promised completeness criterion due to Post [20].