The 4th International Conference on Appications of Computer Systems – ACS'97, Poland, 1997, pp.287-300

INFORMATION ESTIMATION FOR LOGIC FUNCTIONS

Dan Simovici[1)], Vlad Shmerko[2)], Vasily Cheushev[3)], and Svetlana Yanushkevich[2)]

[1)] Dept. of Mathematics & Computer Sciences, University of Massachusetts, Boston MA 02125, USA, Fax: 617 731 3297 E-mail: dsim@cs.umb.edu
[2)] Institute of Computer Science & Information Systems, Technical University, Zolnierska Str. 49, 71210 Szczecin, POLAND, Fax: +4891-4876439, E-mail: shmerko@beta.ii.tuniv.szczecin.pl
[3)] Dept. of Programming Technology, State University, Skorina Ave. 4, 200050, Minsk, BELARUS, E-mail: sasha@cacedu.unibel.by

Abstract. Information theory methods allow us to establish non-traditional approaches to solve some problems of logic design. We make an attempt in this paper to explain a technique to compute information estimations for logic function. The fundamental definitions of information theory within the interpretation of algebra of logic are implied to be base for the information estimations. Special features of computations of the estimations for completely and incompletely specified logic functions, and symmetric logic functions are investigated. A classification for logic functions based on information estimations is introduced. Some results are of a discussion character, since the ways to compute the information estimations for logic functions concerns sometimes the philosophical aspects of information theory.

Key words: *information theoretic approach, information estimations, entropy, logic design, Boolean function, Multi-Valued Logic (MVL) function*

I. INTRODUCTION

An information theoretic approach to solve *logic design* problems attracts specialists' attention two last decades, at least. For instance, enough simple estimations for complexity of a logic function, characteristics of mutual relation between values of function and its arguments, can be obtained by using this approach. In other cases, the information theoretic approach can be an original alternative to traditional methods and approaches, in particular, when solving logic tasks through decision trees.

Among the successful applications of mutual information approach to decision tree design reported already are medical diagnosis, an expert system design tool, character recognition, classification of chess endgames, Neural Networks (NN) learning, various control algorithms based on unification of methods of fuzzy logic and information theory.

Some of the earlier work [Gana73], [Hart82] deals with the conversion of decision tables to decision trees. Goodman and Smyth [Good88] proposed a general top-down mutual information algorithm to design decision trees from deterministic decision tables. It requires that all the probabilities are known a priori.

A natural step is to use this result for minimization of logic functions. Because the truth table of logic function is a special case of decision table, the approach presented in [Hart82] can be applied to the logic function minimization problem. It was shown in [Kaba90] that a minimal sum-of-products form for a Boolean function is solved by entropy evaluation. Authors of [Llor93a,b] have generalized the results achieved in [Kaba90]. They have been proposed for approximate minimization of MVL functions.

The information theoretic approach was applied to solve other problems of Logic Design too. So, for instance, the results on the relationship between entropy and the average power consumption of the circuits generated from

[2), 4)] This work was partially supported by Committee for Scientific Researches, Poland, and by Fund of Fundamental Researches (National Academy of Sciences) and Informatization Fund, Republic of Belarus

Boolean functions were considered in [Chi97]. The entropy measure was shown to provide an effective power estimate for single-output and fully-correlated multiple-output Boolean functions.

A method to determine the entropy of large logic circuits with a level of accuracy was proposed in [Lioy97]. Authors partition the set of output signals according to the information about the functional correlation that may exist among such signals, and then compute the approximate entropy values after performing output clustering. In [Var82] an information theoretic approach for the generation of near-optimum sequential fault location experiments is proposed. Authors consider a general multiple input-output circuits.

There are known many papers where the information theoretic approach is used in NN and fuzzy logic. Methods of fuzzy logic and solution of logic optimization problem on NN is of great interest last time. That is why will show briefly how the information theoretic approach combines the concept of *fuzzy entropy* and NN.

In the designing of NN training procedure information estimations are commonly used as error criterion. It has been suggested to use the information theoretical criterion rather than the least squared error criterion for training NN. For example, Hinton [Hin89] suggest to minimize the cross-entropy. NN compute activation levels or weights of interconnections between processing elements. With proper normalization, these weights range from 0 to 1 and thus represent fuzzy set. Fuzzy entropy can be computed from weights of interconnections between processing elements achieved after every n training cycles. Since NN training is the evolution of a dynamic system from a random set of weights to a set of weights that achieve a certain fit, fuzzy entropy of the weights changes evolves to either a fixed point, a limit cycle, or strange attractor of a chaotic system. Thus, fuzzy entropy provides a basis for optimization of the training of NN, independence of testing net performance against a set of test data samples. A NN that evolves to a fixed point has memorized the training examples; a NN whose weights have maximum entropy has optimum generalized capability.

In [Simo93] it was shown that unlike the notation of entropy of a probability distribution, the entropy of function has an algebraic rather than a probabilistic character. It means that a function (including a logic one) have so called *functional entropy* as a numerical characteristic, satisfying the group of axioms. Authors relay that such approach will allow us to obtain limitations on functional decomposition that are difficult to prove by traditional methods.

Finally, we would like to mention an original papers [Sol95] and [Pav90]. It has to stimulate specialists to find new non-traditional ways to solve logic design problems. The paper [Sol95] study the fundamental question of how query learning performs in imperfectly learnable problems, where the student can learn to approximate the teacher only. It has been proved those queries for minimum entropy in student space (i.e. maximum information gain) lead to the same improvement in generalization performance as for a noisy linear teacher. The efficiency of query learning is thus determined by the structure of the student space alone, and it is supposed that this result holds more generally for minimum student space entropy queries in imperfectly learnable problems.

In paper [Pav90] study the problem how to increase the information density encoded in well known bar codes. For this goal authors have used for the first time information measures. The idea based on that a distinction exists between the method of ”painting” bits on paper (channel encoding) and that of encoding information into bits (source encoding). Information content of a bar code allow to develop new types of bar codes.

The analysis of the mentioned papers allows us to conclude that the technique to compute information estimations for logic function was not formed completely, that is reasoned by the difficulties of the computations concerning sometimes the *philosophical aspects* of information theory. There are also some contradictions in the interpretations of results on the information measure.

Moreover, on our view, there are no papers of a general character, which would be assessable for wide circle of logic designers. Note that the information approach (within the frameworks of application for logic design problems) should be classified as *inter-disciplinary* approach. It means that it is based on new methods, which are *non-trivial unification* of results from various subject areas. A problem of a *mutual interpretation* of the results becomes the actual, for this case - the results from theoretical-applied branches of logic design and information theory. However inter-disciplinary approaches have always been fruitful and resulted in principally new results and understanding the investigated objects.

What could be expected from the application of the information approach? Say, the well-known problem of comparison of optimization algorithms for logic functions is solved by their testing on benchmarks. The strategies of the compared algorithms are evaluated relatively only, i.e. by the result of the testing. In the contrast, the information approach allows for evaluating each of the strategies through information estimations. Additionally, each step of the strategy can be analyzed and compared with a best one (in accordance to the information estimations). It should be especially noted one of the main features of the strategy based on the information measures. It consists in the possibility of a *prognosis* for a next step for the searching. Thus, the information approach is a way to deeply understand and improve the investigated strategy. On the other hand, the algorithms to transform logic functions on the base of the information approach can be an analogue or extension of the class of benchmark function to test other algorithms.

Thus, it can be concluded that it is presently necessary to conceptually explain the methods of the information measures for logic functions, to learn their properties from the position of information approach, to investigate transforms of logic functions as information objects. We suppose that such results could stimulate discussions and researches on application of information methods in logic design. The mentioned is the motivation of this paper.

We are based here on the fundamental papers on information theory [Shan48a,b], [Marti81], [Yag73], and others, and also on the mentioned papers of applied character. The proposed technique to compute information estimation for logic functions includes rules to compute entropy and information quantity for completely and incompletely specified logic function, and for symmetric functions, These rules lie on our theoretical investigations.

Our results, however, go beyond this. We demonstrate that logic functions divided into *classes* from the position of information approach. Such new classification can be efficiently used in logic recognition. We also demonstrate for some simple tasks of logic design how to use the information estimations to reduce the computational capacity.

II. FUNDAMENTAL DEFINITIONS OF INFORMATION THEORY WITHIN THE INTERPRETATION OF ALGEBRA OF LOGIC

2.1. GENERAL STATEMENTS

The subject of our investigations are logic (Boolean and MVL) functions, which will be interpreted as multi-input one-output information objects. We will base below on a number of fundamental statements of information theory and algebra of logic. Let us formulate our premises by the following five statements.

STATEMENT 1. Combinations of variables values and logic function values corresponding to these combinations are considered from the information theory point of view in the sense of *probability distribution* for various mentioned values. It means that information estimation, for instance, for variables values {0, 1, 0, 0} and {1, 0, 1, 1} of Boolean functions *are not differed*.

STATEMENT 2. Assigning the unspecified values for incompletely specified logic function *does not influenced* onto information estimation since the probability of realization for the assigned combination of variables values and logic function values is equal to 0. In other words, the assigning for unspecified values, if it necessary for the given task, can be made by an arbitrary way and does not influenced onto information estimation

STATEMENT 3. A value of logic function or any its variable carries *a proper information* which can be evaluated as $-\log p$, where p is a probability of realization of this value. The base of the *log* can be chosen arbitrary, however we recommend to choose the radix of the number system (binary for Boolean function, m-valued for m-valued logic function), for that information estimation belongs to a normalized range from 0 to 1 (bits or m-ary units accordingly).

STATEMENT 4. The dependence of logic function on an arbitrary variable can be expressed by the term *mutual information*. In other words, we will consider an average mutual information in the given combination of n variables values and a value of logic function.

STATEMENT 5. Any functional transform of a logic function *does not increase the information* about function values for the given combinations of variables values. So, for instance, when transforming logic function given by truth table, into any other form does not increase the information about values of logic function and variables.

2.2. TERMINOLOGY AND ASSIGNMENTS

The following terms establish the relationship between terms of the proposed approach and some notations of information theory.

- Term *a value of logic variable* (also a *value of logic function*) is associated with *an event* in probabilistic theory and with the term a *message* in information theory.
- Term *a set of logic variable values* (or *logic function values*) is related to the term an *ensemble, or a set of possible values with the given probability distribution of these values.*

The notations below are introduced to define *proper information, relative information, mutual information, entropy and conditional entropy* for logic functions.

f - m-valued logic function $f(X) = f(x_1, x_2,.., x_n)$ of n variables $x_1, x_2,.., x_n$;
$f, x_i \in \{0,\dots, m-1\}$, $i = 1,..., n$

k - the number of combinations $U_0, U_1, ..., U_k$ of variables $x_1, x_2,.., x_n$ for which the logic function f is specified; for completely specified logic functions $k = m^n$

$U_j = \{x_1,$ - j-th combination of variables $x_1, x_2,.., x_n$ of the truth table logic function f;

x_2,.., x_n} j = 0,1, ..., k

b - denotes the random variable associated with values of logic function f

$p(b)$ - probability of that the logic function f takes a value $b \in \{0, ..., m\text{-}1\}$

t_i - denotes the random variable associated with values of variable x_i of logic function f

$p(t_i)$ - *probability* of that variable x_i takes a value $t_i \in \{0, ..., m\text{-}1\}$

$I(b)$ - *proper information* in value b of logic function f

$I(t_I)$ - *proper information* in value t_i of variable x_i

$p(b \mid t_i)$ - *conditional probability* of value b of logic function f given value t_i of variable x_i

$I(b \mid t_i)$ - *relative information* in value b logic function f given value t_i of variable x_i

$I(b\,;t_i)$ - *mutual information of jointly specified* values b of logic function f and value t_i of variable x_i

$H(f)$ - *entropy* of logic function f as a measure of uncertainty of function value

$H(f \mid t_i)$ - *conditional entropy* of logic function f given value t_i of variable x_i, i.e. the uncertainty of value of logic function f when value of variable x_i, is given

$H(f \mid x_i)$ - *conditional entropy* of logic function f given variable x_i, i.e. average uncertainty of values of logic function given variable x_i

$H(f x_i)$ - *entropy of jointly specified* logic function f and variable x_i

$I(f\,;x_i)$ - *mutual information* in variable x_i about logic function f and vice versa, that reflects the influence of variable x_i on the function f and conversely, the influence of function f on variable x_i

2.3. INFORMATION IN LOGIC FUNCTION AND ITS VARIABLES

The main terms and expressions related with notations such as *proper information*, *relative information* and *mutual information* for a logic function and its variables are introduced in this Section.

Let us define the notation for a *probability* of values of variable x_i logic function f and the function itself. As it was shown in section 2.2, values of variable x_i is associated with a random variable t_i. Values of the function are associated with random variable b. Both the m-valued logic function f and its variable x_i take values from the range {0,…, m - 1}. Assume, a logic function f is defined for k combinations of variables U_1, U_2, ..., U_k . Accordingly, each variable x_i of the function also takes values for these k combinations.

Definition 2.1. The probability of value t_i of variable x_i is the relation: <*the number of combinations for which this variable takes value* t_i>/< *total number k of the specified combinations*>.

Definition 2.2. The probability of value b of a logic function f is the relation: <*the number of combinations of variables for which the function takes value* b> / <*the number of k of specified combinations*>.

Example 2.1. In accordance with definitions 2.1 and 2.2, the following is true for the Boolean function given by Table 2.1: (i) probability p of value $t_i = 0$ of variable x_i is equal to $p\,(x_i = 0) = {}^3/_5$, and $p\,(x_i = 1) = {}^2/_5$; (ii) the probabilities of values of the Boolean function f take the values $p(f = 0) = {}^4/_5$, and $p(f = 1) = {}^1/_5$.

TABLE 2.1
A fragment of truth table for an incompletely specified Boolean function f

…	x_i	…	f
	0		0
	1		0
	0		0
	1		1
	0		0

Definition 2.3. *Proper information* $I(t_i)$ (or *proper information quantity*) of the value t_i for a m-valued variable x_i of a logic function f be called *the information being carried by the value* t_i:

$$I(t_i) = -\log(p(t_i)), \tag{2.1}$$

where $p(t_i)$ denotes the probability that the variable x_i takes value t_i.

Example 2.2. Proper information for the Boolean function given by Table 2.1 takes the following values: $I(x_i = 0) = -\log_2 {}^3/_5 = 0.737$ bit; $I(x_i = 1) = -\log_2 {}^2/_5 = 1.322$ bit.

Unless otherwise specified we assume that all logarithms are in base m, where m denotes the number of possible values of variable x_i (function f).

By analogy, *proper information* $I(b)$ (or *proper information quantity*) of the value b of a m-valued logic function f be called *the information carried by value* b *of this function*:

$$I(b) = -\log p(b), \tag{2.2}$$

where $p(b)$ denotes the probability of that the function f takes the value b.

Example 2.3. For Boolean function (Table 2.1): $I(f = 0) = -\log_2 {}^4/_5 = 0.322$ bit; $I(f = 1) = -\log_2 {}^1/_5 = 2.322$ bit.

For a fixed value t_i of variable x_i, the probability $p(b \mid t_i)$ of that logic function f takes value b is called *conditional probability* of value b and calculate by follows:

$$p(\hat{a} \mid t_i) = \frac{p(\hat{a},t_i)}{p(t_i)} \quad , \tag{2.3}$$

where $p\,(\boldsymbol{b}, t_i)$ denotes joint probability of values $\boldsymbol{b}$ and t_i.

Example 2.4. For the Boolean function given by Table 2.1, the following probabilities were obtained: $p\,(f = 0, x_i = 0) = {}^3/_5$; $p\,(f = 0, x_i = 1) = {}^1/_5$; $p\,(f = 1, x_i = 0) = 0$ and $p\,(f = 1, x_i = 1) = {}^1/_5$. Then $p\,(f = 0| x_i = 0) = {}^3/_5 : {}^3/_5 = 1$; $p\,(f = 0| x_i = 1) = {}^1/_5 : {}^2/_5 = {}^1/_2$; $p\,(f = 1| x_i = 0) = 0$; $p\,(f = 1| x_i = 1) = {}^1/_5 : {}^2/_5 = {}^1/_2$.

Let the value t_i of variable x_i be given and the above conditional probability distribution is known.

Definition 2.4. The *relative information* $I\,(\boldsymbol{b}| t_i)$ in the value $\boldsymbol{b}$ of logic function f given the value t_i of variable x_i be

$$I\,(\boldsymbol{b}| t_i) = -\log(p\,(\boldsymbol{b}| t_i)) . \tag{2.4}$$

By analogy, the *relative information* $I\,(t_i | \boldsymbol{b})$ in the value t_i of variable x_i given the value $\boldsymbol{b}$ of logic function f be

$$I\,(t_i | \boldsymbol{b}) = -\log p\,(t_i | \boldsymbol{b}) . \tag{2.5}$$

Example 2.5. For Boolean function (Table 2.1) obtain: $I\,(f = 0| x_i = 0) = -\log_2 1 = 0$; $I\,(f = 0| x_i = 1) = -\log_2 {}^1/_2 = 1$ bit; $I\,(f = 1| x_i = 0) = 0$; $I\,(f = 1| x_i = 1) = -\log_2 {}^1/_2 = 1$ bit. Note that for combinations with probability being equal to 0, for instance $(f = 1, x_i = 0)$, we suppose the relative information be equal to 0.

Definition 2.5. The *mutual information* between the values t_i and $\boldsymbol{b}$ is the information in value t_i of variables x_i about the value $\boldsymbol{b}$ of logic function f (or the information in value $\boldsymbol{b}$ about value t_i).

Above mutual information can be calculated by

$$\boldsymbol{I}(\boldsymbol{b}; t_i) = \boldsymbol{I}(f) - \boldsymbol{I}(\boldsymbol{b}| t_i) = -\log p\,(\boldsymbol{b}) + \log p\,(\boldsymbol{b}| t_i) = \log \frac{p(\hat{a}\,|\,t_i)}{p(\hat{a})} . \tag{2.6}$$

Using expression (2.3) for conditional probability, we can write (2.6) as follows

$$I(\hat{a}; t_i) = \log \frac{p(\hat{a}; t_i)}{p(p; t_i)} . \tag{2.7}$$

Mutual information between values t_i of variable x_i and logic function f can be used as measure of *mutual influence* of function value to variable value and vice versa.

Example 2.6. For Boolean function (Table 2.1) obtain: $I\,(x_i = 0| f = 0) = -\log_2 1 = 0$; $I\,(x_i = 0| f = 1) = 0$; $I\,(x_i = 1| f = 0) = -\log_2 {}^1/_2 = 1$ bit; $I\,(x_i = 1| f = 1) = -\log_2 {}^1/_2 = 1$ bit.

Definition 2.6. *Mutual information* between variable x_i and logic function f is the information in variable x_i about the function f and describe by

$$I(f; x_i) = \sum_{b=0}^{m-1} \sum_{t_i=0}^{m-1} p(\hat{a}, t_i) \log \frac{p(\hat{a}\,|\,t_i)}{p(\hat{a})} . \tag{2.8}$$

Using expression (2.3) for conditional probability, write

$$I(f; x_i) = \sum_{b=0}^{m-1} \sum_{t_i=0}^{m-1} p(\hat{a}, t_i) \log \frac{p(\hat{a}, t_i)}{p(\hat{a})\,p(t_i)} . \tag{2.9}$$

Relation (2.9) is preferable from the computational point of view.

Example 2.7. For Boolean function (Table 2.1) obtain: $I\,(f\,;\,x_i) = I\,(x_i\,;\,f) = 0.6\cdot 0.322 - 0.2\cdot 0.678 + 0 + 0.2\cdot 1.322 = 0.322$ bit. It mean that variable x_i carries 0.322 bits of information about function f, and vice versa, function f carried 0.322 bits of information about variable x_i .

Mutual information between variable and logic function is used below as measure of influence of logic function on its variable and vice versa. Next Section is devoted to the estimations which allow to measure a range of *uncertainty* of values for logic function and its variables.

2.4 ENTROPY AND CONDITIONAL ENTROPY OF LOGIC FUNCTION AND ITS VARIABLES

As a rule, to estimate the uncertainty of values of random numbers, entropy and conditional entropy of them is necessary. Consider the using these estimations for logic function and its variables.

Definition 2.7. *Shannon's entropy* of variable x_i of logic function f is *the average proper information* of values t_i of this variable and written by

$$H(x_i) = \sum_{t_i=0}^{m-1} p(t_i) I(t_i) = -\sum_{t_i=0}^{m-1} p(t_i) \log p(t_i) . \tag{2.10}$$

By analogy, Shannon's entropy of logic function f is *the average proper information* of values f of this function and defined as follows:

$$H(f) = \sum_{t_i=0}^{m-1} p(\hat{a}) I(\hat{a}) = -\sum_{t_i=0}^{m-1} p(\hat{a}) \log p(\hat{a}) . \tag{2.11}$$

Example 2.8. For the Boolean function given by Table 2.1, the entropy of variable x_i defined by $H\,(x_i) = -{}^3/_5 \cdot \log_2 {}^3/_5 - {}^3/_5 \cdot \log_2 {}^3/_5 = 0.971$ bit. The entropy of function f defined by $H\,(f) = -{}^4/_5 \cdot \log_2 {}^4/_5 - {}^1/_5 \cdot \log_2 {}^1/_5 = 0.722$ bit.

It is naturally to ask: are the considered estimations for logic functions grounded sufficiently? Furthermore, is it correct from the mathematical point of view to use Shannon's entropy for logic functions? It is known another approach based on a notation of functional entropy [Simo93]. Let us consider the relationship between Shannon's entropy and functional entropy for logic functions.

Since function f takes some values corresponding to k combinations of variables only, the number of various probabilities $p(\boldsymbol{b})$ can take no more than k. Let r_b denotes the number of combinations of variables values for which logic function takes value $\boldsymbol{b}$. The probability $p(\boldsymbol{b})$ is expressed by $p(\boldsymbol{b}) = r_b / k$, where $\boldsymbol{b} \in \{0, \ldots, m\text{-}1\}$, $r_b \in \{0, \ldots, k\}$. Therefore, Shannon's entropy of logic function f (2.11) can be written by

$$H(f) = -\sum_{b=0}^{m-1} \frac{r_b}{k} \cdot \log \frac{r_b}{k}. \tag{2.12}$$

Expression (2.12) is transformed as follows

$$H(f) = \log k - \frac{1}{k} \sum_{b=0}^{m-1} r_b \cdot \log r_b. \tag{2.13}$$

Definition 2.8. *Functional entropy* of logic function f defined as [Simo93]

$$h(f) = \sum_{b=0}^{m-1} r_b \cdot \log r_b \tag{2.14}$$

We use in (2.14) designation $h(f)$ to distinct functional entropy from Shannon's entropy (2.11).
So, the functional entropy represents a numerical characteristic of a logic function f.

Taking to account (2.14) we can formulate the relationship between Shannon's and functional entropy.

Affirmation 2.1. Shannon's entropy of logic function f is related to functional entropy as

$$H(f) = \log k - \frac{1}{k} h(f). \tag{2.15}$$

Proof follows from equations (2.12 -2.14).

We don't investigate here relationship (2.15) in detail and its philosophical sense. Note only, that this relationship seems to allow for explaining some surprising results obtained sometimes for information measures and minimization of logic functions basing Shannon's entropy. All mentioned above allow us to consider some properties of entropy and mutual information of logic function and its variables. Unless otherwise specified, we refers below to Shannon's entropy.

The following properties are useful to exploit to compute the information estimations for logic functions.

Properties

- The entropy $H(f)$ of logic function f takes values from the range $0 \le H(f) \le 1$.
- The entropy $H(x_i)$ of any variable x_i of logic function f belongs to the range $0 \le H(x_i) \le 1$.
- The entropy $H(c)$ of constant logic function f is equal to 0, i.e. $H(const) = 0$.
- The entropy $H(x_i)$ of any variable x_i of completely specified logic function f is equal 1.

  Proof. A completely specified logic function is specified for m^n combinations of variables, where n is the number of variables of the function. The probability of value $t_i \in \{0,\ldots, m\text{-}1\}$ of variable x_i is equal to $p(t_i) = {}^1/_m$. Taking into account (2.10), we obtain

  $$H(x_i) = -\sum_{t_i=0}^{m-1} {}^1/_m \cdot \log_m {}^1/_m = 1 \text{ bit}. \qquad \textbf{Q.E.D.}$$

- The entropy of some statistical independent logic functions is equal to sum of the entropy of ones.
- Mutual information between incompletely specified logic function f and variable x_i which takes constant value only, is equal to 0.

Proof. Let $x_i = b$ denotes the fact that variable x_i takes constant value b only. We can write $p(t_i) = 1$ for $t = b$ and $p(t) = 0$ for any $t \neq b$. Since $p(\boldsymbol{b}|t_i) = p(\boldsymbol{b})$ when t is fixed to constant, then $\log(p(\boldsymbol{b}|t_i)/p(\boldsymbol{b})) = 0$. Therefore, it follows from (2.8) that $I(f, x_i) = 0$ bit given $x_i = const$. **Q.E.D.**

Definition 2.9. Given value t_i of variable x_i the *conditional entropy* $H(f|t_i)$ of logic function f be a measure of average uncertainty of logic function value with respect to the value t_i of variable x_i and calculated by:

$$H(f \mid t_i) = \sum_{b=0}^{m-1} p(\hat{a},t_i) I(\hat{a} \mid t_i) = -\sum_{b=0}^{m-1} p(\hat{a},t_i) \log \frac{p(\hat{a})}{p(t_i)} \tag{2.16}$$

Definition 2.10. Given variables x_i, the *conditional entropy* $H(f|x_i)$ of logic function f be a measure of average uncertainty of logic function values with respect to the variable x_i

$$H(f|x_i) = -\sum_{b=0}^{m-1} p(\boldsymbol{b}) \; H(\boldsymbol{b}|t_i) = -\sum_{b=0}^{m-1} \sum_{t_i=0}^{m-1} p(\boldsymbol{b}, t_i) \; \log \; p(\boldsymbol{b}|t_i). \tag{2.17}$$

Example 2.9. For Boolean function (Table 2.1) the following estimations were obtained: $H(f\,|\,x_i = 0) = p(f = 0, x_i = 0) \cdot I(f = 0\,|\,x_i = 0) + p(f = 1, x_i = 0) \cdot I(f = 1\,|\,x_i = 0) = {}^3/_5 \cdot 0 + 0$. It means that when variable x_i takes value 0, then value of the function is pre-defined: $H(f\,|\,x_i = 1) = p(f = 0, x_i = 1) \cdot I(f = 0\,|\,x_i = 1) + p(f = 1, x_i = 1) \cdot I(f = 1\,|\,x_i = 1) = {}^1/_5 \cdot 1 + {}^1/_5 \cdot 1 = 0.4$ bit. $H(f\,|\,x_i = H(f\,|\,x_i = 0) + H(f\,|\,x_i = 1) = 0.4$ bit. So, value of variable x_i reduces the average uncertainty of values of the function from $H(f) = 0.722$ bit to $H(f\,|\,x_i) = 0.4$ bit.

Definition 2.11. The entropy $H(fx_i)$ of jointly specified variable x_i and logic function f is defined by

$$H(fx_i) = -\sum_{b=0}^{m-1}\sum_{t_i=0}^{m-1} p(\boldsymbol{b}, t_i)\; I(\boldsymbol{b}, t_i) = -\sum_{b=0}^{m-1}\sum_{t_i=0}^{m-1} p(\boldsymbol{b}, t_i)\; \log\; p(\boldsymbol{b}, t_i)\,. \tag{2.18}$$

Since $p(\boldsymbol{b}, t_i) = p(t_i\,|\,\boldsymbol{b}) \cdot p(\boldsymbol{b})$ we have

$$H(fx_i) = -\sum_{b=0}^{m-1}\sum_{t_i=0}^{m-1} p(\boldsymbol{b}, t_i)\; \log\,(p(t_i\,|\,\boldsymbol{b}) \times p(\boldsymbol{b})) =$$

$$= -\sum_{b=0}^{m-1}\sum_{t_i=0}^{m-1} p(\boldsymbol{b}, t_i)\; \log p(t_i\,|\,\boldsymbol{b}) - \sum_{b=0}^{m-1}\sum_{t_i=0}^{m-1} p(\boldsymbol{b}, t_i)\; \log p(\boldsymbol{b})$$

It means that the entropy is additive on respect of logic function f and variable x_i

$$H(fx_i) = H(f) + H(x_i\,|\,f) \quad \text{or} \tag{2.19}$$

$$H(fx_i) = H(x_i) + H(f\,|\,x_i)\,. \tag{2.20}$$

Here we have used the property $p(\boldsymbol{b}, t_i) = p(t_i)\; p(\boldsymbol{b}\,|\,t_i)$.

Example 2.10. Compute the conditional entropy and entropy of jointly specified function and variable for the Boolean function (Table 2.1).We have obtained that $H(f) = 0.722$ bit, $H(x_i) = 0.971$ bit, and $H(f\,|\,x_i) = 0.4$ bit. The entropy of joint specified Boolean function f and variable x_i is calculated in accordance with (2.18) as follows: $H(fx_i) = {}^3/_5 \cdot \log_2 {}^3/_5 + {}^1/_5 \cdot \log_2 {}^1/_5 + {}^1/_5 \cdot \log_2 {}^1/_5 = 1.371$ bit. Accordingly with (2.20), $H(fx_i) = 0.971 + 0.4 = 1.371$ bit.

2.5 MUTUAL INFORMATION AND ENTROPY FOR LOGIC FUNCTION AND SUBSET OF ITS VARIABLES

Now we generalize these results for logic function and group of variables.
Let $\{x_{i_1}, x_{i_2}, \dots, x_{i_q}\}$ is subset of logic variables $x_{i_s} \in \{x_1, x_2, \dots, x_n\}$, $s = 1, 2, \dots, q$; $q = 1, 2, \dots, n$; $i_s \in 1, 2, \dots, n$. Notation $x_{i_1}x_{i_2}\dots x_{i_q}$ be interpreted as *joint assignment* of elements of the sub-sets (ensembles) $x_{i_1}, x_{i_2}, \dots, x_{i_q}$. Denote the joint assignment of logic function f and components of the sub-set $x_{i_1}, x_{i_2}, \dots, x_{i_q}$ as $fx_{i_1}x_{i_2}\dots x_{i_q}$ (the joint assignment of ensemble f and discrete ensembles $x_{i_1}, x_{i_2}, \dots, x_{i_q}$).

Lemma 2.1. Given joint specified subset $\{x_{i_1}, x_{i_2}, \dots, x_{i_q}\}$ of variables and logic function f, the entropy defined by

$$H(fx_{i_1}x_{i_2}\dots x_{i_q}) = H(f) + H(x_{i_1}\,|\,f) + H(x_{i_2}\,|\,fx_{i_1}) + \dots + H(x_{i_q}\,|\,fx_{i_1}x_{i_2}\dots x_{i_{q-1}})\,. \tag{2.21}$$

Proof. For any values $t_{i_1}, \dots, t_{i_q}$ of variables $x_{i_1}, x_{i_2}, \dots, x_{i_q}$ and value $\boldsymbol{b}$ of logic function f the joint probability is equal to $p(\boldsymbol{b}, t_{i_1}, \dots, t_{i_q}) = p(\boldsymbol{b}) \cdot p(t_{i_1}\,|\,\boldsymbol{b}) \cdot p(t_{i_2}\,|\,\boldsymbol{b}\,t_{i_1}) \cdot \dots \cdot p(t_{i_q}\,|\,\boldsymbol{b}\,t_{i_1}\dots t_{i_{q-1}})$, where $p(t_{i_s}\,|\,\boldsymbol{b}\,t_{i_1}\dots t_{i_{s-1}}) = p(\boldsymbol{b}, t_{i_1}, \dots, t_{i_s}) / p(\boldsymbol{b}, t_{i_1}, \dots, t_{i_{s-1}})$. Then $I(\boldsymbol{b}; t_{i_1}; \dots; t_{i_q}) = I(\boldsymbol{b}) + I(t_{i_1}\,|\,\boldsymbol{b}) + I(t_{i_2}\,|\,\boldsymbol{b}\,t_{i_1}) + \dots + I(t_{i_q}\,|\,\boldsymbol{b}\,t_{i_1}\dots t_{i_{q-1}})$, where $I(\boldsymbol{b}; t_{i_1}, \dots, t_{i_q}) = -\log\,p(\boldsymbol{b}, t_{i_1}, \dots, t_{i_q})$, $I(t_{i_s}\,|\,\boldsymbol{b}\,t_{i_1}\dots t_{i_{s-1}}) = -\log p(t_{i_s}\,|\,\boldsymbol{b}\,t_{i_1}\dots t_{i_{s-1}})$. Compute average of $I(\cdot)$ and obtain (2.21).

Q.E.D.

Lemma 2. 2. The entropy for jointly specified subset $\{x_{i_1}, x_{i_2}, \dots, x_{i_q}\}$ of variables and logic function f is expressed by

$$H(fx_{i_1}x_{i_2}\dots x_{i_q}) = H(x_{i_q}) + H(x_{i_{q-1}}\,|\,x_{i_q}) + H(x_{i_{q-2}}\,|\,x_{i_{q-1}}x_{i_q}) + \dots + H(x_{i_1}\,|\,x_{i_2}\dots x_{i_q}) + H(f\,|\,x_{i_1}x_{i_2}\dots x_{i_q}) =$$

$$= \sum_{j=1}^{q} H(x_{i_s}\,|\,x_{i_{s-1}}\dots x_{i_1}) + H(f\,|\,x_{i_q}\dots x_{i_1})\,. \tag{2.22}$$

It is proved by the way similar to proof of *Lemma 2.1*.

The properties (2.21) and (2.22) shown that entropy of logic function is *additive*. It is of great important significance for information estimations of logic functions. We will explain below a recurrent technique to compute these estimations on decision trees, that is applied, in particular, to minimize logic functions. The following theorem is of important for the mentioned technique.

Theorem 2.1. Given subset $\{x_{i_1}, x_{i_2}, \dots, x_{i_q}\}$ of variables, the conditional entropy of logic function f with respect to these variables defined by

$$H(f\,|\,x_{i_1}x_{i_2}\dots x_{i_q}) = H(fx_{i_1}x_{i_2}\dots x_{i_q}) - H(x_{i_1}x_{i_2}\dots x_{i_q})\,. \tag{2.23}$$

Proof. For the pair f, x_i in accordance with (2.20) write $H(f|x_i) = H(fx_i) - H(x_i)$. For the subset $x_{i_1} x_{i_2}$ of variables and logic function f we have $H(fx_{i_1} x_{i_2}) = H(x_{i_1} x_{i_2}) + H(f|x_{i_1} x_{i_2})$. It means that $H(f|x_{i_1} x_{i_2}) = H(fx_{i_1} x_{i_2}) - H(x_{i_1} x_{i_2})$. Then, for the subset of three variables and logic function f write by induction $H(f|x_{i_1} x_{i_2} x_{i_3}) = H(f\, x_{i_1} x_{i_2} x_{i_3}) - H(x_{i_1} x_{i_2} x_{i_3})$. Thus, generalizing for subset $\{x_{i_1} x_{i_2} \ldots x_{i_q}\}$ and logic function *f*, we have the desired result (2.23). **Q.E.D.**

Now, let us consider the relations between entropy and mutual information of logic functions. Since $p(\boldsymbol{b}, t_i) = p(t_i)\ \ p(\boldsymbol{b}|t_i)$, we can rewrite the expression (2.9) for mutual information as follows

$$I(f; x_i) = -\sum_{b=0}^{m-1} \sum_{t_i=0}^{m-1} p(\boldsymbol{b}, t_i) \log \frac{p(\hat{a}\,|\,t_i)}{p(\hat{a})} =$$

$$= -\sum_{b=0}^{m-1} \sum_{t_i=0}^{m-1} p(\boldsymbol{b}, t_i) \log p(\boldsymbol{b}|\,t_i) + \sum_{b=0}^{m-1} \sum_{t_i=0}^{m-1} p(\boldsymbol{b}, t_i) \log p(\boldsymbol{b}).$$

Therefore mutual information

$$I(f; x_i) = H(f) - H(f|x_i) \tag{2.24}$$

Using $p(\boldsymbol{b}, t_i) = p(t_i\,|\,\boldsymbol{b}) \cdot p(\boldsymbol{b})$ we have

$$I(f; x_i) = H(x_i) - H(x_i|f) \tag{2.25}$$

Thus, mutual information between logic function *f* and variable x_i can be presented in entropy terms.

Let us generalize expression (2.25) for a group of variables of a logic function.

Theorem 2.2. Mutual information between a subset $\{x_{i_1}, x_{i_2}, \ldots, x_{i_q}\}$ of variables and logic function *f* defined by

$$I(f; x_{i_1} x_{i_2} \ldots x_{i_q}) = H(f) + H(x_{i_1} x_{i_2} \ldots x_{i_q}) - H(f\, x_{i_1} x_{i_2} \ldots x_{i_q}). \tag{2.26}$$

Proof. To proof the desired expression we use the previous theorem. Taking account (2.23) for *q*=1 write $I(f; x_{i_1}) = H(f) - H(f|x_{i_1}) = H(f) + H(x_{i_1}) - H(fx_{i_1})$. Generalize the equation above for variables $x_{i_1} x_{i_2}$ obtain $I(f; x_{i_1} x_{i_2}) = H(f) - H(f|x_{i_1} x_{i_2}) = H(f) + H(x_{i_1} x_{i_2}) - H(fx_{i_1} x_{i_2})$. Thus, generalizing for subset $\{x_{i_1}, x_{i_2}, \ldots, x_{i_q}\}$ and logic function *f*, we have the desired result.

Q.E.D.

Note that both the estimations (entropy and information) are of importance and evaluated in sections below.

III. INFORMATION ESTIMATIONS FOR LOGIC FUNCTIONS

3.1. INFORMATION ESTIMATIONS FOR LOGIC FUNCTIONS

Let a logic function f of n variables is completely specified for all combinations of variables values by its truth table. Discuss the notation of logic function from information theory point of view.

We will consider the following information estimations (Table 3.1):

- *An influence* of variable x_i on logic function f, or mutual information $I(f; x_i)$ in x_i about f. Such estimation in evaluated by expression (2.24).
- *An influence* of a group $x_{i_1}, x_{i_2}, \ldots, x_{i_q}$ of variables on logic function *f*. The mutual information $I(f; x_{i_1} x_{i_2} \ldots x_{i_q})$ is calculated by (2.26).
- *An average uncertainty* of values of logic function *f* without taking into account information about values of arguments of the function, or its entropy $\acute{I}(f)$ computed by (2.11),
- *An average uncertainty* of values of logic function *f* taking into account the known value of variable x_i, i.e. the conditional entropy $H(f|x_i)$ computed by (2.16) and (2.23).
- *An average uncertainty* of values of logic function *f* taking into account the known value of the group of variables $x_{i_1}, x_{i_2}, \ldots, x_{i_q}$; so, the conditional entropy $H(f|x_{i_1} x_{i_2} \ldots x_{i_q})$ is evaluated by expression (2.23).

TABLE 3.1: Main information estimation for logic function

Estimation	Calculation	Means
$I(f; x_i)$ *Mutual information* between variable x_i and function f	$H(f) - H(f\|x_i)$	An influence of variable x_i on logic function f
$I(f; x_{i_1} x_{i_2} \ldots x_{i_q})$ *Mutual information* between the function and group of variables	$H(f) + H(x_{i_1} x_{i_2} \ldots x_{i_q}) - H(f\, x_{i_1} x_{i_2} \ldots x_{i_q})$	An influence of a group $x_{i_1}, x_{i_2}, \ldots, x_{i_q}$ of variables on logic function *f*.

$Í(f)$ *Entropy* of logic function f	$-\sum_{b=0}^{m-1} p(b)\cdot\log p(b)$	An average uncertainty of logic function values
$H(f\,\vert\,x_i)$ *Conditional entropy* of logic function with respect to variable x_i	$H(fx_i) - H(x_i)$	An average uncertainty of logic function values given value of variable x_i
$H(f\,\vert\,x_{i_1}x_{i_2}\ldots x_{i_q})$ *Conditional entropy* of logic function with respect to group of variables	$H(fx_{i_1}x_{i_2}\ldots x_{i_q})$ - $H(x_{i_1}x_{i_2}\ldots x_{i_q})$	An average uncertainty of logic function values given group of variables $x_{i_1}, x_{i_2}, \ldots, x_{i_q}$

Let us consider how to calculate the above estimations for Boolean functions of two variables. Note that we consider a logic function as an information object. It means that it can be said about information estimations of a Boolean function. Really, variables x_1 and x_2 are initial functions, and a logic operation over the arguments allows us to obtain a new function.

Example 3.1. Compute the information estimations of the Boolean function f (m=2) of two variables. Note, that radix of logarithms is the radix of Boolean algebra, i.e. 2.

The entropy $H(f)$ of logic function f in accordance with (2.11) is defined by $H(f) = -p(f=0)\cdot\log_2 p(f=0)$ - $p(f=1)\cdot\log_2 p(f=1)$. Here and below probability $p(f=0)$ is defined as the number of combinations of variables of the function, for which the function takes value 0, divided by the total number of the combination. Probability $p(f=1)$ and others computed adequately.

- The entropy $H(x_i)$ of variable x_i, (i = 1, 2) of logic function f in accordance with (2.10) is expressed as $H(x_i) = -p(x_i=0)\cdot\log_2 p(x_i=0) - p(x_i=1)\cdot\log_2 p(x_i=1)$.
- The entropy of the joint specified variable x_i and Boolean function f is defined by (2.18) $H(fx_i) = -p(x_i=0, f=0)\cdot\log_2 p(x_i=0, f=0) - p(x_i=0, f=1)\cdot\log_2 p(x_i=0, f=1) -$ $p(x_i=1, f=0)\cdot\log_2 p(x_i=1, f=0) - p(x_i=1, f=1)\cdot\log_2 p(x_i=1, f=1)$;
- The conditional entropy $H(f\,\vert\,x_i)$ of logic function f value with respect to the variable x_i can be calculated in accordance with (2.23) as $H(f\,\vert\,x_i) = H(fx_i) - H(x_i)$.
- Mutual information in variable x_i about logic function f in accordance with (2.24) is computed by $I(f; x_i) = H(f) - H(f\vert x_i)$, or, in accordance with (2.26) $I(f; x_i) = H(f) + H(x_{i_1}) - H(fx_i)$.

Let us apply the expounded technique to compute information estimations for Boolean functions of two variables (Table.3.2). Consider in detail how such estimations are evaluated for Boolean functions, in particular, for AND function.

Example 3.2. Calculate the information estimations for AND function of two variables.

(i) The entropy of variables: $H(x_1) = H(x_2) = -{}^2/_4\cdot\log_2 {}^2/_4 - {}^2/_4\cdot\log_2 {}^2/_4 = 1$ bit. Really, the number of combinations of variables of the function, for which variable x_1 (x_2) takes value 0, is equal to 2. Total number of the combinations is 4. Therefore the probability $p(x_1=0) = p(x_2=0) = {}^2/_4$. Likely, $p(x_1=1)$ and $p(x_2=1)$.

(ii) The entropy of function: $H(f) = -{}^3/_4\cdot\log_2 {}^3/_4 - {}^1/_4\cdot\log_2 {}^1/_4 = 0.81$ bit. Here the probability $p(f=0)$ of zero value of the function is equal to ${}^3/_4$, since the function takes value 0 for 3 from 4 combinations of variables, and $p(f=1) = {}^1/_4$, since the function is equal to 1 for one combination only.

(iii) The probability that the function f takes the value 0 and the variable x_1 (or x_2) takes the value 0, i.e. the probability p (0, 0) of the combination (f = 0, x = 0) is equal to ${}^1/_4 + {}^1/_4 = {}^2/_4$; $p(0, 1) = {}^1/_4$; $p(1, 0) = 0$; $p(1, 1) = {}^1/_4$. Thus the entropy $H(fx)$ in accordance with (2.18) is calculated as $H(fx) = -{}^2/_4\cdot\log_2 {}^2/_4 - {}^1/_4\cdot\log_2 {}^1/_4 - 0 - {}^1/_4\cdot\log_2 {}^1/_4 = 1.5$ bit.

TABLE 3.2: Information estimations for Boolean functions of 2 variables

	Entropy			Mutual information	
	$H(f)$	$H(f\|x_1)$	$H(f\|x_2)$	$I(f;x_1)$	$I(f;x_2)$
const 0	0	0	0	0	0
x_1x_2	0.81	0.5	0.5	0.31	0.31
$x_1 \neg x_2$	0.81	0.5	0.5	0.31	0.31
x_1	1	0	1	1	0
$\neg x_1 x_2$	0.81	0.5	0.5	0.31	0.31
x_2	1	0.5	0	0	1
$x_1 \oplus x_2$	1	1	1	0	0
$x_1 + x_2$	0.81	0.5	0.5	0.31	0.31
Inverse symmetry of Boolean functions					
$x_1 \downarrow x_2$	0.81	0.5	0.5	0.31	0.31
$x_1 \sim x_2$	1	1	1	0	0
$\neg x_2$	1	1	0	0	1
$x_1 \to x_2$	0.81	0.5	0.5	0.31	0.31
$\neg x_1$	1	0	1	1	0
$x_2 \to x_1$	0.81	0.5	0.5	0.31	0.31
$x_1 \| x_2$	0.81	0.5	0.5	0.31	0.31
const 1	0	0	0	0	0

Note, that the entropy here is more that 1, since the talk is about the jointly given ensembles, and the total number for these unified events is equal to 4.

(iv) Conditional entropy $H(f|x)$, i.e average uncertainty of function value given x (x_1 or x_2): using (2.23) we have $H(f|x) = H(fx) - H(x) = 1.5 - 1 = 0.5$ bit.

(v) Mutual information $I(f;x)$ between the variable x_1 (or x_2) and logic function f in accordance with (2.24): $I(f;x) = 0.81 - 0.5 = 0.31$ bit or, in accordance with (2.26) it is equal to $I(f;x) = 0.81 + 1 - 1.5 = 0.31$ bit.

Observation 3.1. The adequate information estimations has the Boolean function $x_1 | x_2$, since its truth vector $\mathbf{X} = [1110]$ is connected with the truth vector is related to truth vector $\mathbf{X} = [0001]$ of function AND through the inversion (Table 3.2). This property is generalized for Boolean functions of n variables.

Property 3.1. Boolean functions are of the same information estimations if they takes the inverted values for the same combinations of their variables.

3.2. CLASSES OF BOOLEAN FUNCTIONS OF TWO VARIABLES ON RESPECT TO INFORMATION CRITERIA

From the position of information approach, logic functions have another, different from traditional classification. This classification is based on information quantity and estimation of information affect of variables on the function. For Boolean function of two variables these estimations are given in Table 3.2. Note, that the uncertainty of variables is maximal, i.e. $H(x_1) = H(x_2) = 1$ bit for completely specified functions. The mutual influence of variables is equal to 0, and also $H(x_1|x_2) = 1$ and $I(x_1|x_2) = 0$. It is coherent with one of central statements of information theory.

Within the frameworks of the traditional classification, say, Post classification, each of these functions belongs to one of five classes. It is followed from Table 3.2, that Boolean functions of two variables can be divided onto four groups (classes) of Boolean functions of two variables:

- class of *0-information* functions,
- class of functions *saving* information,
- class of functions with information *dependence on one of variables*,
- class of functions *reducing* information.

Notation *0-information* for a Boolean function means that (i) uncertainty of values of the function is equal to 0, (ii) a value of the function does not depend on values of variables, i.e. there is no dependency of variables on function and vice versa.

Notation *a function saving information* means that the value of the function for any combination of variables values depends on values of *both the arguments* for the combination, and vice versa - each variable affects the function by all its values. Hence, a value of one variable can be restored when the value of other variable and the function are known.

Conception *information dependence of a logic function on one of variables* denotes the fact that values of the function depend on one variable only, and the influence of other variable onto the function is equal to 0.

Notation *function reducing information* means, that value of the function is defined by both the arguments for a group of variables values combinations only. For some combinations the function depends on one variable only. Such logic function *partially* saves information about variables. Hence, to define values of the second variables by the unique way, it is not enough to know value of the function and value of one of variables.

Thus, the classification of Boolean functions of two variables from the information position allows us to conclude: (i) there are four classes of Boolean functions of two variables, and (ii) the logic functions which belong to one class, have the same information estimations, i.e. they are not differed on the information estimation inside each of the classes. Note, that such the classification produces ideas to construct new types of *classifiers* for logic functions (features). The properties of the information estimations are saved when increasing the number of variables [Smer98].

3.3. INFORMATION ESTIMATIONS FOR MVL FUNCTIONS

Consider some features of computation of the information estimations for MVL functions. Here and blow we will use the logarithm of radix *m*. We are based on that the number of digits to represent the numbers in an *m*-valued algebra, is defined by the mentioned logarithm. Since there are now essential differences between the computation of the information estimations for Boolean and MVL functions, let us concern the results on evaluating the influence of the radix of the function on the information estimations.

Example 3.3. Compute the information estimations 3-valued logic function $(x_1+x_2)_{mod\ 3}$ (Table 3.3).

- The entropy of 3-valued function *f* in accordance with (2.11) we can write $= - p(f=0) \cdot \log_3 p(f=0) - p(f=1) \cdot \log_3 p(f=1) - p(f=2) \cdot \log_3 p(f=2)$.
- The entropy $H(fx)$ of joint specified the variable *x* (x_1 or x_2) and 3-valued function *f* $H(fx) = p(f=0, x=0) \cdot \log_3 p(f=0, x=0) - \ldots - p(f=2, x=2) \cdot \log_3 p(f=2, x=2)$.
- Conditional entropy $H(f \mid x)$ of logic function *f* value with respect to the value t_i of variable x_i can be calculated in accordance with (2.20) as $H(f \mid x) = H(fx) - H(x)$.
- Conditional entropy $H(x \mid f)$ of variable *x* (x_1 or x_2) with respect to 3-valued function *f* in accordance with (2.19) we can write $H(x \mid f) = H(fx) - H(f)$.
- Mutual information between the variables x_i and 3-valued logic function *f* in accordance with (24) we can write $I(f; x_1) = H(f) - H(f \mid x)$ or, with accordance with (2.26) $I(f; x_1) = H(f) + H(x) - H(fx)$.

The results on evaluating the information estimations for this and other logic function (Table 3.4) are given in Table 3.4.

Observation 3.2. As result of operation $(x+y)_{mod\ m}$, a class of logic functions is formed with the same information estimations independent on *m*.

TABLE 3.3: Modulo *m* (*m*=3, 4), GF(4) addition and multiplication operators

$(x_1+x_2)_{mod\ 3}$

	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

$(x_1+x_2)_{mod4}$

	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

$(x_1x_2)_{mod3}$

	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

$(x_1x_2)_{mod4}$

	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

Addition over GF(4)

	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1

Multiplication over GF(4)

	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1

TABLE 3.4: Information Estimations of operations of Addition and Multiplication of 3- and 4-valued logic functions two variables

f	$H(f)$	$H(fx)$	$H(f \mid x)$	$H(x \mid f)$	$I(f;x)$
$(x+y)_{mod3}$	1	2	1	1	0
$(x+y)_{mod4}$	1	2	1	1	0
Addition GF(4)	1	2	1	1	0
$(xy)_{mod3}$	0.906	1.665	0.665	0.759	0.241
$(xy)_{mod4}$	0.876	1.626	0.626	0.750	0.250
$(xy)_{mod5}$	0.957	1.800	0.800	0.843	0.157
Multiplication GF(4)	0.939	1.750	0.750	0.811	0.189

3.4 INFORMATION ESTIMATIONS OF SYMMETRIC BOOLEAN FUNCTIONS

Symmetric Boolean functions are the subject for especial attention in logic design. That is why it is interest to obtain their information estimations.

Remind that a symmetric function *f*(*X*) on *n* binary variables can be represented by a binary string of length *n*+1, $f_0\ f_1 \dots f_n$, where f_i is the value of function *f*(*X*) when number *i* of the *n* variables are 1. From the positions of the information approach, the class of symmetric logic function is characterized by *that the influence of each of arguments on function is the same*. It allows us to essentially reduce the computations for the information estimations.

Let us demonstrate the calculation of condition entropy of the Boolean function given one of arguments for AND function of 3 and 4 variables. We show below that the technique of computations is much more easy for symmetric Boolean functions. Moreover, we propose some rules for the computation which are suitable for synthesis the algorithm. We will use the equation $H(f|x) = H(fx) - H(x)$ to compute the conditional entropy, taking into account that *H*(*x*) for a completely specified function is equal to 1.

TABLE 3.5: On computing the information estimations symmetric Boolean functions OR, and of 3- and 4 variables

3-variables Boolean functions					
$p(x=0)$	$^1/_8$	$^2/_8$	$^1/_8$	0	
$p(x=1)$	0	$^1/_8$	$^2/_8$	$^1/_8$	
$p(f_i)$	$^1/_8$	$^3/_8$	$^3/_8$	$^1/_8$	
f_i	**0**	**1**	**2**	**3**	
OR	0	1	1	1	
AND	0	0	0	1	
4-variables Boolean functions					
$p(x=0)$	$^1/_{16}$	$^3/_{16}$	$^3/_{16}$	$^1/_{16}$	0
$p(x=1)$	0	$^1/_{16}$	$^3/_{16}$	$^3/_{16}$	$^1/_{16}$
$p(f_i)$	$^1/_{16}$	$^4/_{16}$		$^4/_{16}$	$^1/_{16}$
f_i	**0**	**1**	**2**	**3**	**4**
OR	0	1	1	1	1
AND	0	0	0	0	1

Example 3.4. Compute the information estimations for Boolean AND function of 3 variables. Apply the following form to express a symmetric Boolean function (Table 3.5)

- The entropy $H(f)$ of Boolean function *f*: the probability $p(0)$ that logic function *f* takes the value 0 is equal to $p(0) = {}^1/_8 + {}^3/_8 + {}^3/_8 = {}^7/_8$; $p(1) = {}^1/_8$. $H(f) = -{}^7/_8 \cdot \log_2 {}^7/_8 - {}^1/_8 \cdot \log_2 {}^1/_8 = 0.544$ bit.
- The entropy $H(fx)$ of joint specified Boolean function *f* and variable *x*: the probability $p(0,0)$ of the combination $(f=0, x=0)$ is equal to $p(0,0) = {}^1/_8 + {}^2/_8 + {}^1/_8 = {}^4/_8$; the probability of the combination $(f=0, x=1)$ is $p(0,1) = {}^1/_8 + {}^2/_8 = {}^3/_8$; the probabilities $p(1,0) = 0$; $p(1,1) = {}^1/_8$ are computed by analogy. The entropy $H(fx) = -{}^4/_8 \cdot \log_2 {}^4/_8 - {}^3/_8 \cdot \log_2 {}^3/_8 - 0 - {}^1/_8 \cdot \log_2 {}^1/_8 = 1.406$ bit
- Conditional entropy is equal to $H(f|x) = H(fx) - H(x) = 1.406 - 1 = 0.406$ bit.
- Mutual information $I(f;x)$ between the logic function *f* and a variable *x* is equal to $I(f;x) = H(f) - H(f|x) = 0.544 - 0.406 = 0.138$ bit or $I(f;x) = H(f) + H(x) - H(fx) = 0.544 + 1 - 1.406 = 0.138$ bit.

Let us analyze the symmetric Boolean functions of two variables through their information estimations and extract the classes of symmetric functions in accordance with the estimations. Like Boolean functions, symmetric logic function are grouped into. *Dependent on one of variables when the influence of each of arguments on to function is the same*

The result obtained can also be generalized for symmetric Boolean functions *n* variables

Affirmation 3.1. Any symmetric Boolean functions of *n* variables belongs to one of three information classes: (i) *0 - information*, (ii) *Saving information*, and (iii) *Reducing information*.

The *Proof* follows from the definition of the symmetric functions and a structure of the classes for 2-variable Boolean function.

3.5 INFORMATION ESTIMATION FOR INCOMPLETELY SPECIFIED LOGIC FUNCTIONS

The problem to manipulate by incompletely specified logic function is very actual in logic design. Traditionally, incompletely specified logic functions are used to derive the flexibility (non-observability, non-controllability) at a node of a circuit. In a number of cases, the problem can be solved by the trivial assignment of values for the initial data. However, this approach is suitable for logic functions of small number of variables. For strongly unspecified logic function, the *don't care about don't cares* principle.

Evaluating the information estimations for incompletely specified logic function is based on the fact that the probabilities for assigning *don't cares* are equal to 0. That is way (i) it is necessary to analyze logic function for the given values of variables; (ii) the assigning the unspecified values does not change the information about the function.

Example 3.5. Compute the information estimations for an incompletely specified Boolean function of 3 variables given for 4 combinations (Table 3.6) .

The values entropy $H(x_I)$, conditional entropy $H(f|x)$, mutual information $I(f;x)$, mutual information $I(x_I;x_1)$ and mutual information $I(x_I;x_3)$ are given in Table.3.7.

TABLE 3.6: Incompletely specified Boolean function *f* of 3 variables

x_1	x_2	x_3	f
0	1	1	1
1	0	0	1
1	1	0	0
0	1	0	0

TABLE 3.7: Information estimations for an incompletely specified Boolean function (Table 3.6)

	Entropy		Mutual information			
	$H(x_I)$	$H(f\|x)$	$I(f;x)$	$I(x_I;x_1)$	$I(x_I;x_2)$	$I(x_I;x_3)$
x_1	1	0	0	0	0.31	0.31
x_2	0.81	0.69	0.5	0.31	0	0.12

IV. DISCUSSION AND CONCLUSION

The interest on using information theory methods was paid comparatively recently to solve problems of logic design. These methods are principally differed from the traditional ones and require to change the point of view on many classical conceptions of logic functions and their manipulations. The information approach makes wider our understanding, but its application is connected with some difficulties.

The information approach seems to be promising to solve the problem of comparison of optimization algorithms for logic functions by evaluating each of the strategies through information estimations. An unique possibility of the information approach to form a *prognosis* for the searching algorithms is not studied yet. Applying this property can strongly affect on the searching algorithms. A classification of Boolean functions by information estimations allows for new possibilities for logic recognition (logic functions are very specific objects, for which it is difficult to draw parallels on recognition, for instance, symbols and speech based on information approach). There are known attempts to apply information approach to investigate decomposability of logic function.

Anyway, the information approach is a way to understand deeply various manipulations with logic functions which are the base for many methods and algorithms of logic design.

Our main objective is to attract specialists' attention to information approach to solve logic design problems.

In this paper we systematically explain the ways to adapt and interpret the information theory methods to solve logic design problems. We propose a computational technique for information estimations of logic functions and their manipulations. This technique is oriented onto wide circle of specialists. We illustrate various algorithmic, computational aspects by a number of examples.

We stand on the position that a logic functions has got information which can be estimated numerically. The methods of the information theory allow us for such estimations. For the same time, the logic function has got many specific features in contrast to classical objects of the information theory. We have showed here how to interpret various information estimations for logic functions, what computational features have symmetric Boolean functions got, how to classify logic functions from the positions of information approach.

We have systematized and developed the ways to compute information estimations for some typical methods of description of logic functions (completely and incompletely specified Boolean and MVL functions, symmetric functions). By other words, we propose the technique to compute information estimations for logic functions which is simple to understand and apply.

The approach outlined in this paper is a basis for further work. In particular, we are investigating the problem of information estimations for strategies of logic functions minimization.

REFERENCES

Chi-Hong Hwang, and A.C.-H.Wu (1997), An Entropy Measure for Power Estimation of Boolean Functions, *Proc. of the ASP-DAC'97 - Asia and South Pacific Design Automation Conference,* Japan, pp.101-106

Ganapathy S., and Rajaraman V.(1973), Information Theory Applied to the Conversion of Decision Tables to Computer Programs, *Commun. of the ACM*, vol. 16, pp. 532 - 539.

Goodman R.M., and Smyth P. (1988), Decision Tree Design from a Communication Theory Standpoint, *IEEE Trans. on Information Theory*, vol. 34, no. 5, pp. 979 - 994.

Hartmann C.R.P., Varshney P.K., Mehrotra K.G., and Gerberich C.L. (1982), Application of Information Theory to the Construction of Efficient Decision Trees, *IEEE Trans. on Inf. Th.*, vol. IT-28, pp.565-577

Hinton G.E.(1989), Connectionist Learning Procedures, *Artifical Inelligence,* no.40, pp.185-234

Kabakcioglu A.M., Varshney P.K., and Hartman C.R.P. (1990), Application of Information Theory to Switching Function Minimization, *IEE Proceedings*, Pt E, vol.137, pp.389-393

Lioy A., Macli E., Poncino M., and Rossello M. (1997), Accurate Entropy Calculation for Large Logic Circuits Based on Output Clustering, *Proc. Great Lakes Symp. on VLSI*, Urbana-Champaign, IL, USA, pp.70-75

Lloris-Ruiz A., Gomez-Lopera J.F., and Roman-Roldan R. (1993a), Entropic Minimization of Multiple-Valued Functions, *Proc. 23rd Int. Symp. on Multiple-Valued Logic*, pp.24-28

Lloris A., Gomez J.F., and Roman R. (1993b), Using Decision Trees for the Minimization of Multiple-Valued Functions, *Int. J. Electronics*, vol.75, no.6, pp.1035-1041

Martin N.F.G., and England J.W.(1981), Mathematical Theory of Entropy *Addison-Wesley, Reading, MA*

Pavlidis T., Swartz J., and Wang J.P. (1990), Fundamentals of Bar Code Information Theory, *IEEE Computer*, April, pp.74-86

Shannon C.E. (1948a), A Mathematical Theory of Communication, Part 1, *Bell Syst. Tech.J.*, vol.27, pp.379-423

Shannon C.E. (1948b), A Mathematical Theory of Communication, Part 1, *Bell Syst. Tech.J.*, vol.27, pp.623-656

Shmerko V., Cheushev V., and Yanushkevich S. Classification of Logic Functions based on Information Approach. *Automatics and Telemechanics, Russian Academy of Sciences* (in Russian), Translated: *Automation Remote Control* - to be published in Russian and in English

Simovici D.A. and Reischer C. (1993), On Functional Entropy, *Proc. ISMVL'93*, pp.100-104

Sollich P. (1995), Minimum Entropy Queries for Linear Students Learning Nonlinear Rules, *Proc. of the 3rd European Symp. on Artificial Neural Networks*, pp.217-222

Varshney P.K., Hartmann C.R.P., and De Faria J.M. (1982), Application of Information Theory to Sequential Fault Diagnosis, *IEEE Trans. on Computers*, vol. C-31, pp.164-170

Yaglom I.M. (1973), *Probability and Information*, Moscow (In Russian)