IJBSCHS(2012-18-01-12)
[Original article]
Biomedical Soft Computing and Human Sciences, Vol.18,No.1,pp.69-76

(Accepted on 2011.03.28)

An Algebraic Method to Identify Classes of Formulas in Calculus of Predicates

Vasyl SEMERENKO*

*Department of Computer Technique, National Technical University, Vinnytsia, UKRAINE

Abstract: *The P-complete problem of logic deduction in classical propositional and first order predicate calculus is considered. The suggested method is alternative to the traditional methods of direct and inverse logic deduction in particular to the Robinson's resolution principle. The deterministic algorithm of polynomial complexity based on Boolean algebra of cubic functions for differentiation between two classes of formulas: valid and satisfiable, or unsatisfiable and satisfiable, is suggested.*
Keywords: *Logic deduction, Resolution principle, Propositional calculus, Predicate calculus, Formula, Boolean algebra, NP-complete problem.*

1. Introduction

One of the major directions in Artificial Intelligence is solution of the problem of knowledge representation and its processing based on the logic deduction [1]. Intellectual methods for selection and generalization of information allow one to make motivated decisions and carry out the long-term predictions in economics, medicine, and other areas.

The problems of logic deduction, as well as the majority of problems for the optimum solution search in various spheres can be classified as NP-complete problems. One way to solve such problems is to use special algorithms, for example, genetic algorithms [2], the Fuzzy Adaptive Search Method [3], the DNA computing [4], etc. However, these methods give only approximated solution and do not guarantee finding the global optimum. More effective solution of this problem means to reduce it to P-complete problem and to design the exact algorithms of polynomial complexity for them.

Because of the increase of the initial data dimensions, both the NP-complete and P-complete problems, require a lot of computing resources, that is why the development of algorithms suitable for parallel processing is also important.

95, Khmelnytske shose
VNTU, Vinnytsia, 21021, Ukraine
Office Phone: +38-0432-43-90-02
e-mail: vpsemerenko@mail.ru

2. The mathematical tools for formalization of logic deduction

The procedure of forming the new knowledge from the already available one in any formal system of knowledge representation is a logic deduction which can be written as follows:

$$A_1 \rightarrow (A_2 \rightarrow \dots (A_n \rightarrow C) \dots), \qquad (1)$$

where $A_1, A_2, \dots, A_n$ – premises of logic deduction, C – conclusion of logic deduction.

The theoretical basis of Artificial Intelligence is mathematical logic the main results of which are stated in the famous monograph by D. Hilbert and P. Bernays [5].

The deduction offered by D. Hilbert and based on the set axioms is a difficult and time consuming process, because the choice of the necessary axioms at each step of the formula proof remains at the level of guess-work, intuitions, and finally leads to an exhaustive search.

It is also difficult to automate a natural deduction which uses heuristics to select the rules.

In 60s the new stage began in the development of mathematical logic [6]. The methods of logic deduction which have developed in that period can be divided into two groups. The methods of direct logic deduction perform the proof from premises to conclusion, and the methods of inverse logic deduction operate in the reverse direction, beginning from the conclusion, through a chain of rules for search of the facts confirming the proof. The most well-known method of inverse logic deduction is the Robinson's resolution principle [7].

The main disadvantage of both approaches is caused by the necessity to execute a lot of search operations while performing the proof from the premises to the conclusion or vice versa. To decrease the quantity of superfluous operations it is necessary to use complicated heuristics.

More effective problem solution of logic deduction can be found by reducing the problem to decision procedure. In modern logic the decision procedure is understood as the task of finding general methods for identification of the validity or unsatisfiability of logic formulas [8, 9]. Checking of the correctness of some reasoning which is preset by the premises $A_1, A_2, \dots, A_n$ and by the conclusion C is logically equivalent to the proof of the fact that the formula

$$\overline{A_1} \vee \overline{A_2} \vee \dots \vee \overline{A_n} \vee C \qquad (2)$$

is valid, or the formula

$$A_1 \,\&\, A_2 \,\&\, \dots \,\&\, A_n \,\&\, \overline{C} \qquad (3)$$

is unsatisfiable (inconsistent).

We will consider the classical propositional and first order predicate calculus as the formal theories in which we will investigate the problems of validity and unsatisfiability.

The research problem of formula deductibility in propositional calculus can be replaced by the equivalent research problem of the functions validity in Boolean algebra of logic functions. While performing propositional calculus we transform formula $F(A_1, A_2, \dots, A_n)$ into the logic function $f(A_1, A_2, \dots, A_n)$, and then formulas (2) and (3) can be represented in the disjunctive normal form (DNF) or conjunctive normal form (CNF) of the logic function, in which premise A_i represents the i th clause of DNF or CNF. In this case, the decision procedure consists in identification of the class of logic function $f(A_1, A_2, \dots, A_n)$:

1) the class of the valid functions accepting the value of "true" in all sets of values of their arguments;
2) the class of the unsatisfiable functions accepting the value of "false" in all sets of values of their arguments;
3) the class of the satisfiable functions accepting the value of "true" at least on one value set of their arguments;

By analogy to propositional calculus, formulas of the first order predicate calculus are divided also into three classes: valid, unsatisfiable and satisfiable formulas.

Effective hardware-software implementation of a decision procedure can be reached by means of the Boolean algebra of cubic functions.

The Boolean algebra of cubic functions is isomorphic to the Boolean algebra of logic functions [10]. In the Boolean algebra of logic functions, the operations of conjunction, disjunction and negation correspond to the operations of intersection, union and complement of cubes in the Boolean algebra of cubic functions, and the normal forms correspond to cubic coverings [11] .The cubic K-covering (cubic Q-covering) of some logic function f is the representation of CNF of direct function f (inverse function $\overline{f}$) in cubic form (i.e. in alphabet $\{0,1,x\}$).

One CNF clause corresponds to one cube of a covering, the direct (inverse) value of a variable in CNF clause corresponds to the value one (value zero) of cube components, otherwise cube component is equal to the value of "x", and the number m of Boolean function variables is equal to the number m of cube components.

The cubic D-covering (cubic R-covering) of some logic function f is the representation of CNF of direct function f (inverse function $\overline{f}$) in cubic form. The conformity between the DNF clauses and the cubes of D-covering (cubic R-covering) is the same, as the one for K-covering (cubic Q-covering).

If the cube d_i contains r components equal to the value of "x" then we will assume, that the cube d_i has an interval r ($r = 0 \div m$) and m-interval cube we designate as $X_m = \{x\,x \dots x\}$.

The cubic covering is called as minimal if there is no other covering of this type with smaller number of the cubes.

For example, CNF of a logic function f and the inverse function $\overline{f}$,

$$f(a,b,c,d) = (a \vee \overline{b} \vee c)\,\&\,(b \vee \overline{c})\,\&\,(\overline{a} \vee c \vee d)$$

$$\overline{f}(a,b,c,d) = (a \vee b \vee c)\,\&\,(\overline{a} \vee c \vee \overline{d})\,\&\,(\overline{b} \vee \overline{c})$$

correspond to the following coverings

$$K = \begin{vmatrix} 1 & 0 & 1 & x \\ x & 1 & 0 & x \\ 0 & x & 1 & 1 \end{vmatrix};$$

$$Q = \begin{vmatrix} 1 & 1 & 1 & x \\ 0 & x & 1 & 0 \\ x & 0 & 0 & x \end{vmatrix}.$$

The DNF of logic function f and inverse function $\overline{f}$,

$$f(a,b,c,d) = \overline{a}\,\&\,\overline{b}\,\&\,\overline{c} \vee a\,\&\,\overline{c}\,\&\,d \vee b\,\&\,c$$

$$\overline{f}(a,b,c,d) = \overline{a}\,\&\,b\,\&\,\overline{c} \vee \overline{b}\,\&\,c \vee a\,\&\,\overline{c}\,\&\,\overline{d}$$

correspond to the following coverings

$$D = \begin{vmatrix} 0 & 0 & 0 & x \\ 1 & x & 0 & 1 \\ x & 1 & 1 & x \end{vmatrix}; \qquad R = \begin{vmatrix} 0 & 1 & 0 & x \\ x & 0 & 1 & x \\ 1 & x & 0 & 0 \end{vmatrix}.$$

If the initial data for the task of a logic deduction is presented in the form of the traditional set of the premises and conclusion, they will correspond to the clauses of CNF and DNF or the cubes of K-covering and R-covering.

Intersection operation of the cube $d_i = \{d_{i1},\ldots,d_{iz},\ldots,d_{im}\}$ and of the cube $d_j = \{d_{j1},\ldots,d_{jz},\ldots,d_{jm}\}$ is designated as $d_k = d_i \cap d_j$, and it leads to the cube $d_k = \{d_{k1},\ldots,d_{kz},\ldots,d_{km}\}$, which is the common part of cubes d_i and d_j. This operation is carried out in two stages: at the beginning the preliminary cube d_k'' is being formed according to Table 1, and then the final cube d_k is being formed in the following way: if the cube d_k'' does not contain a component equal to value of "y" then the final cube d_k coincides with the cube d_k'', otherwise the result of intersection operation will be empty.

Tab.1. Formation of the components d_{kz} of the preliminary cube d_k''

d_{iz} \ d_{jz}	0	1	x
0	0	y	0
1	y	1	1
x	0	1	x

The $*$-product operation of cubes is also required for us. As a result of $*$-product of cube d_i and cube d_j the cube $d_k = d_i * d_j$ will be produced. It contains separate subcubes of these cubes, while cubes d_i and d_j do not contain any common subcubes. This operation is also carried out in two stages: at the beginning the preliminary cube d_k'' is being formed according to Table 1, and then the final cube d_k is being formed in the following way: if the cube d_k'' contains only one component, equal to the value of "y" then it is replaced by the value of "x", otherwise the result of $*$-product operation will be empty.

By means of cubic coverings it is possible to represent both formulas of propositional calculus and formulas of first order predicate calculus.

3. Decision procedure

in propositional calculus

In the modern theory of logic deduction, the basic method of performing the resolving procedure is the Robinson's resolution principle [7]. The essence of this rule of deduction means consecutive forming of the new clauses from the initial set of CNF clauses till an empty clause is derived. Such result will testify that the function f and the formula (3) corresponding to this CNF are unsatisfiable.

According to the famous theorem of resolution principle, completeness [6] the logic deduction based on this method will be completed successfully only if the set of the clauses (i.e., logic function) is unsatisfiable. If a function f belongs to the class of satisfiable functions then the empty clause cannot be derived, hence, decision procedure based on the Robinson's resolution principle will never be completed.

Let's consider another approach to perform logic deduction by means of constructing decision procedure on the basis of direct proof of validity of formula (2) or of formula (3) unsatisfiability and by using cubic coverings.

THEOREM 1. i) For a valid function presented in form (2) or in form (3), the K-covering and R-covering are empty, and the minimal D-covering and Q-covering are equal to the m-interval cube X_m:

$$D = X_m, \quad Q = X_m, \quad K = \varnothing, \quad R = \varnothing . \qquad (4)$$

ii) For an unsatisfiable function presented in form (2) or in form (3), the D-covering and Q-covering are empty, and the minimal K-covering and R-covering are equal to the m-interval cube X_m:

$$D = \varnothing, \quad Q = \varnothing, \quad K = X_m, \quad R = X_m . \qquad (5)$$

iii) For a satisfiable function presented in form (2) or in form (3), the D-covering, R-covering, K-covering and Q-covering are not empty and are not equal to the m-interval cube X_m.

***Proof*:** First, we will prove the statement of item i). From definition of the valid function f it follows that there are no sets of arguments on which the function is false. Hence, the inverse function $\overline{f}$ is empty and there will be also the empty R-covering which corresponds to it. Such function cannot be presented in the CNF, and its perfect DNF contains 2^n clauses which, as a result of minimization, will be combined in one clause, in which there are no literals, i.e., the function becomes equal to logical 1. Such way of forming cubic coverings gives us equality (4).

The other items of the Theorem are established in an analogous way. ■

The type of Boolean function, as well as the type of formula (1), can be defined by the type of the normal form or by the type of the cubic covering. If certain pairs of coverings are known (D_f and K_f, D_f and R_f, K_f and Q_f, R_f and Q_f) then it is possible to identify the type of formula (1) at once. We will further consider the general case of identification when only one of the specified coverings is known.

The proof of the formulas' validity based on resolution principle reduced to the proof that CNF, which corresponds to logic function f, is absent, and in terminology of cubic coverings it means that the covering K_f is empty. We will show that the opposite way of the proof, namely, the proof that one of coverings equals to cube X_m is more effective.

If the majority of the known proof procedures are the methods of inverse deduction (methods of refutation) then the suggested proof procedure is the method of the direct deduction. The first advantage of the decision procedure based on the direct deduction gives the possibility to differentiate two classes of formulas: valid and satisfiable, or unsatisfiable and satisfiable ones.

The procedure of such a proof in algebra of logic functions is equivalent to the first stage of Boolean functions minimization, i.e. equivalent to the calculation of prime implicants.

From the practical point of view (an ability to conduct the parallel processing [12]) it is better to execute the decision procedure in algebra of cubic functions. We will consider an algorithm of distinguishing between valid and satisfiable functions f with the help of known D_f-covering, which consists of n m-digit cubes d_i ($i = 1 \div n$).

ALGORITHM

1. Form the coverings: $D^{(0)} = D_f$ and $D^{(1)} = \varnothing$. Establish iteration number $p = 0$, and sign $w = 0$.
2. Increase the iteration number p: $p = p + 1$.
3. In the end of $D^{(p)}$-covering, add all cubes of $D^{(p-1)}$- covering. Form the covering $D^{(p+1)} = \varnothing$.
4. Introduce the parameter n_p equal to the number of cubes in the $D^{(p)}$-covering.
5. For $i = 1$ till $(n_p\text{-}1)$, execute the following steps:

5.1. For $j = i + 1$ to n_p do:

5.1.1 Execute the $*$-product operation of cubes $d_k = d_i * d_j$ from a $D^{(p)}$-covering.

5.1.2 If the m-interval cube $d_k = X_m$ is obtained then to pass to step 8.

5.1.3. If the interval of cube d_k exceeds an interval of cubes d_i, d_j or of both cubes, then add the cube d_k into the $D^{(p+1)}$-covering, and delete the initial cubes with the smaller interval from the $D^{(p)}$-covering. Update the sign w: $w = w + 1$.

5.1.4. If the cube d_i was removed then go to step 5.

6. If $w = 0$ then go to step 7, else go to step 2.
7. The function f corresponding to the initial D_f-covering is valid. Go to step 9.
8. The function f corresponding to the initial D_f-covering is satisfiable.
9. Stop.

Similar results can be obtained if the Q_f-covering of function f is used as initial data of the algorithm. When we use K_f-covering or R_f-covering of function f as initial data, then it is possible to distinguish between unsatisfiable and satisfiable functions. We will remind that the Robinson's resolution principle identifies only one class of functions.

In the tasks of logic deduction based on known strategies of direct and reverse deduction, in particular, the resolution principle, in general case an undetermined search is used [9], which makes us to characterize these tasks as NP-complete problems.

In contrast to the resolution principle, which increases the number of the clauses at each step of deduction, the suggested method can be named as algebraic, and it results in a gradual decrease of the quantity of cubic covering cubes.

Let's make an estimation of the computational complexity of suggested algorithm according to the number of the operations $*$-product used. In this case, its upper limit of complexity is equal to $O(n^2 p)$, where n is the number of cubes in the initial covering and p is the number of iterations.

To identify any of the functions of the three classes of functions in classical propositional calculus, we have constructed the deterministic algorithm of polynomial complexity, hence, the deducibility problem based on the identification of the formulas classes belongs to the P-complete problems.

4. Decision procedure in the first order predicate calculus

As it follows from the famous Church's Theorem, there is no universal algorithm allowing to identify a class of any formula of predicate calculus [5]. It means that unlike the propositional calculus, the predicate calculus is generally undecisable except for the one-place predicates calculus. We will search, therefore, a separate decision procedure for the three types of the predicate formula: A type (in all predicates, the terms represent the constants), B type (in all predicates, the terms represent the variables), and C type (in all predicates, the terms can represent both constants and variables).

The predicate formula of A type can be easily replaced by the propositional calculus formula, that is why the algorithm of decision procedure discussed in the above section is completely applicable.

The predicate formula of B type has an infinite subject base. Herbrand [6] offered an approach to the solution of the problem of logic deduction in predicate calculus by means of replacing the checking of formulas in infinite area with the checking of them in Herbrand's base of the finite size.

As a matter of fact, Herbrand's interpretation means transformation from predicate calculus to propositional calculus because each fundamental form is equivalent to the proposition. As a result, CNF of the predicate formula is being converted to CNF of formulas of propositional calculus. The cubic equivalent of such a CNF is K_f-covering, with the number of cubes equal to the number of CNF clauses, and the number of components of a cube is equal to the number of fundamental forms. Therefore, it is also possible to apply the algorithm of decision procedure discussed above to such predicate formulas.

Generally, if Herbrand's base consists of n elements (constants) and the formula for propositional calculus contains m one-place predicates, then there will be n^m fundamental concretized expressions and, thus, the same number of cubic coverings. Therefore, even with the availability of a small number of constants in Herbrand's base, it is more effective to perform the checking of deductibility of the predicate formula by a direct replacement of universal quantifiers and existential quantifiers with the corresponding cubic coverings.

For one-place predicates, it is possible to use a simpler approach which does not require the use of the decision procedure based on Herbrand's base. In formulas with the specified types of predicates, it is possible to remove all quantifiers, to replace variables in predicates with constants, and then to define the correctness of logic deduction according to the rules of propositional calculus.

Let's consider now the cubic interpretation of the predicate formula of C type. Let the predicate formula in clause form contain m one-place predicates, in which the terms can be both variables and constants from the base

$$M = \{a_1, a_2, \ldots, a_n\} . \tag{6}$$

Then its K_f-covering will consist of the left part, corresponding to the constant propositions of the predicate formula, and the right part, corresponding to the variable propositions of the predicate formula. In the left part of the covering, the i th column corresponds to the ith constant from base (6), and the right part of the covering j th column corresponds to j th variable of predicate ($i = 1 \div n \times m$, $j = 1 \div m$). The number of cubes in K_f-covering is equal to the number of CNF clauses, while the value one (value zero) of cube component corresponds to direct (inverse) value of the predicate. It should be noted that a cubic representation of the C type formula is the association of cubic representations of the A type and B type formulas.

While producing a logic deduction for formulas of C type we will use the operation of substitution $\{ a_i / w_j \}$, i.e., of replacement of some variable w_j with a constant a_i from area (6). According to the above-shown representation of predicate formulas, the operation of substitution $\{ a_i / w_j \}$ is performed using intersection operation for the column in the right part of the covering, which column corresponds to the variable predicate $E(w_j)$, with the column in the left part of the covering, which column corresponds to the constant predicate $E(a_i)$.

The result of this operation will be written on the place of column $E(a_i)$, and the column $E(w_j)$ is being removed from the covering. Such process gives the concretized expression of cubic covering, and we will call the covering, which lacks the right part, the completely concretized covering (as a matter of fact, this covering is the covering of the A type formula).

Further on, it is necessary to define the type of the concretized covering and, thus, the type of predicate formula by means of the above-mentioned Theorem and Algorithm.

It is also necessary to note that some additional research for optimization of decision procedure for many-place predicates must be carried out.

For instance, for the predicate formula

$$\forall y \forall z (E(y) \vee G(z)) \,\&\, \overline{E}(a_1) \,\&\, \overline{G}(a_2) ,$$

which is defined on the base $M_{1,2} = \{a_1, a_2\}$, the following covering (the left and right parts of this covering are separated by dotted line) corresponds to:

$$\begin{array}{cccc} E(a_1) & G(a_2) & E(y) & G(z) \end{array}$$

$$K_f = \left| \begin{array}{cc:cc} x & x & 1 & 1 \\ 0 & x & x & x \\ x & 0 & x & x \end{array} \right| . \qquad (7)$$

Producing a logic deduction for a formula of the C type we use the operation of substitution $\{ a_i / w_j \}$, where some variable w_j is being replaced by a constant a_i from the base $M_{1,2}$. For example, let us perform a $\{ a_1 / y \}$ substitution for covering (7). For this reason we perform the intersection operation for $E(a_1)$ and $E(y)$ columns (for convenience, we place the specified columns horizontally)

$$\begin{array}{cccccc} \cap & 1 & x & x & : & E(y) \\ & x & 0 & x & : & E(a_1) \\ \hline & 1 & 0 & x & & \end{array}$$

After executing this operation, we will obtain the following partially concretized covering :

$$\begin{array}{ccc} E(a_1) & G(a_2) & G(z) \end{array}$$

$$K_f = \left| \begin{array}{cc:c} 1 & x & 1 \\ 0 & x & x \\ x & 0 & x \end{array} \right| ,$$

which corresponds to the predicate formula

$$(E(a_1) \vee G(z)) \,\&\, \overline{E}(a_1) \,\&\, \overline{G}(a_2) \,.$$

After calculation of the completely concretized covering, it is necessary to identify the type of the concretized covering and, respectively, the type of the predicate formula using above-mentioned Algorithm.

5. Conclusions

In this paper, we suggest to interpret the problem of a logic deduction as the one of the direct proof validity of formula (2) or unsatisfiability of formula (3). This enables us to distinguish among three classes of formulas in the following pairs: valid and satisfiable formulas, or unsatisfiable (inconsistent) and satisfiable formulas. Since we designed the deterministic algorithm of polynomial complexity to solve this problem, it is possible to consider it as a P-complete problem.

The main difference between the suggested algebraic method of logic deduction and the established methods of direct and reverse logic deduction is that the suggested method does not contain the concepts of “premises” and "conclusion" and directly uses the cubic coverings or Boolean functions.

Application of the Boolean algebra of cubic functions allows us to formalize the procedure of logic deduction and to obtain its parallel implementation at the macro-level (the parallelizing of separate algorithm branches) and at the micro-level (the parallelizing of intersection operation and $*$-product operation for cubic coverings) [12].

The obtained theoretical results are likely to find practical application in medical OLAP systems [13]. Such systems contain traditional database of case histories, and knowledge base which allows to form new knowledge (for example, predict the state of health of a person) by means of logic deduction, fuzzy logic [14], and other approaches.

References

[1] Russell, S. and Norvig P. “Artificial Intelligence: A Modern Approach”, (Third edition), Prentice Hall, 2010.

[2] Gupta, S.M. and Imtanavanich P. “Evolutionary Computational Approach for Disassembly Sequencing in a Multiproduct Environment”, Biomedical Soft Computing and Human Sciences, Vol.15, No.1, pp.73-78., 2010.

[3] Li, Q. and Maeda Y. “Fuzzy Adaptive Search Method for Parallel GA Based on Diversity Measure”, Biomedical Soft Computing and Human Sciences, Vol.13, No.1, 2008.

[4] Abu Bakar, R.B. and Watada J. “A Biologically Inspired Computing Approach to Solve Cluster-Based Determination of Logistic Problem”, Biomedical Soft Computing and Human Sciences, Vol.13, No.2, pp.59-66., 2008.

[5] Hilbert, D. und Bernays P. “Grundlagen der Mathematic”, Vol.1. Springer-Verlag, Berlin, Heidelberg, New York, 1968.

[6] Chang, C.L. and Lee, R.C.T. “Symbolic Logic and Mechanical Theorem Proving”, Academic Press, New York, London, 1973.

[7] Robinson, J.A. “A machine oriented logic based on

the resolution principle", J. ASM, No.12, pp.23-41, 1965.

[8] Fitting, M. "First-Order logic and Automated Theorem Proving", Springer-Verlag, New York, 1990.

[9] Vagin, V.N., Golovina, E. Yu., Zagoryanskay, A.A. and Fomina M.V. "Exact and Plausible Reasoning in Intelligent Systems", Ed. by Vagin, V.N. and Pospelov D.A., Fizmatlit, Moscov, 2004.

[10] Semerenko, V.P. "Systolic implementation of cubic functions", Electronic modeling, No.1, pp.21-25, 1992. (Russian edition).

[11] Miller, R.E. Switching Theory, Vol.1, Wiley, New York, 1965.

[12] Semerenko, V.P. "Parallel implementation of a logic deduction", Mathematical machines and systems, No.2, pp.16-24, 1997. (Russian edition).

[13] Wrembel, R. "Data Warehouses and Olap: Concepts, Architectures and Solutions", IGI Global, 2006.

[14] Castillo, O. and Melin P. "Type-2 Fuzzy Logic: Theory and Application", Springer-Verlag, Berlin, Heidelberg, New York, 2008.

Vasyl SEMERENKO

He received Diploma of Engineer in Polytechnical Institute, Vinnytsia, Ukraine in 1976, and PhD in 1985. He has worked as a researcher, computer science lecturer, Associate Prof. at Department of computer technique in National Technical University, Vinnytsia from 1976 to now. His present research interests include artificial intelligence, error-correcting codes, parallel data processing, failure diagnostics. He is a member of IEEE, Computer Society.