UDK 004.832.38

**M.V. Volchenko**
Donetsk National technical university, Donetsk, Ukraine

# RESOLUTION ALGORITHM OPTIMIZATION USING TERNARY CLAUSES REPRESENTATION

***Annotation***

***Volchenko M.V. Resolution algorithm optimization using ternary clauses representation.*** *The work is devoted to solving the SAT problem of propositional logic formulas, that characterized by a great power. The ternary representation of clauses is proposed. Resolution algorithm using ternary clauses representation is proposed. The experimental results confirming the effectiveness of the proposed algorithm.*

***Key words:****satisfiability solving, propositional logic, clause, resolution algorithm, ternary representation, connection matrix.*

At present deductive algorithms are widely used in the decision-making systems, deductive databases, information retrieval systems. Among other uses of deductive inference can also be called the verification of specifications of computer equipment (deductive procedure used, for example, in the AMD processors verification), validation of security systems, the analysis of the correctness data transfer protocols.

One of the main problems faced by the developers of deductive procedures is an exponential increase of the search space. The problem of making deductive procedures in these ever-growing volumes of data to be processed that can effectively work with large sets of clauses has a particular importance.

Inference procedure should satisfy the following requirements to process efficiently clauses sets of big volume:

– to narrow a search space of pairscontained complementary literals on each step of resolution;

– to exclude clauses that can`t be used in the proof from further consideration;

– an effective algorithm of selecting complementary pairs for resolution has to be implemented in the inference procedure.

One way to improve the process of inference is resolution on graph structures as the basis of constructing deductive procedure. A number of deductive inference algorithms on graph structures have been created. However, practical problems solvingis characterized by an exponential increase of the disjuncts amount. These algorithms don`t always show satisfying results. It led to further research in the improvement of inference procedures.

In this paper, we consider the problem of automating the resolution algorithm in propositional logic based on building relations matrix using ternary clauses representation.

## 1. Problem statement

The Boolean satisfiability problem (SAT) is the seminal NP-complete problem described by S. Cook in 1971. Given a Boolean function $f(v_1,\ v_2, \dots, v_n)$in n variables, the SAT problem is the question if there exists an assignment to the variables $v_1,\ v_2, \dots, v_n$ so that the function evaluates to true, or if no such assignments exists, i.e. *f=false*. In the former case *f* is called satisfiable, in the latter unsatisfiable. A set of variable assignments that satisfies a Boolean expression is called a model.

SAT problems are usually given in conjunctive normal form (CNF), which is a product of sum-terms. Each sum-term or clause is the Boolean OR of a number of literals, which are variables or negated variables. Clauses, which contain only one literal, are called unit-literal clauses.

For a CNF to become satisfied, each clause must be satisfied (i.e. evaluate to true). This is the case if at least one literal in the clause evaluates to true. Unit-literal clauses can only be satisfied if their single literal evaluates to true; this forced assignment is called an implication. The implied assignment must be made in the entire CNF, possibly leading to further implications. Assigning implications until no further implications are present is called Boolean constraint propagation (BCP).

If a variable occurs only in one polarity (negated or not negated), the literal can be assigned true without changing the satisfiability of expressions. This is the pure literal rule.

The connection graph method was designed by R. Kowalski. A connection graph is a scheme for representing the proper first-order formulas in disjunctive normal form. Each literal is associated with a node in the connection graph. Literals in a clause are combined into a group. If the literals in two clauses form a contrary pair (P and ¬P) then there is an edge between the respective nodes of the connection graph.

To prove the unsatisfiability of a clause set we must generate and resolve the initial connection graph, i.e. derive an empty clause.

The main operation in connection graph refutation is the link resolution, when the resolventis computed and added to the graph. The corresponding link is deleted and the links of the added resolvent are inserted. A pure clause is a literal group containing a node with no links. Pure clause with all its links can be easily removed from the graph without losing the completeness of the connection graph refutation procedure. Similarly, if we have a tautology clause, it also can be removed from a graph. If a resolvent on some step is a pure clause or a tautology, there is no need to insert this clause into a graph.

The refutation algorithm consists of the following steps.

1. the verification whether there is any clause in a graph or not. If there are no clauses, the algorithm
2. terminates unsuccessfully. If there is the empty clause, then the algorithm is successfully terminated,
3. else go to step 2;
2. if a graph does not contain any link, then the algorithm is unsuccessfully terminated, else go to step 3;
3. a link selection. The link is resolved and the resolvent is generated;
4. if an empty resolvent is obtained, then the algorithm terminates successfully, else the resolvent is inserted into the graph, its links are added, and the algorithm goes to step 2.

The fundamental problem in the connection graph refutation is the choice of suitable links by some criteria at each step of an algorithmic operation. Links are usually selected by using heuristics. In this paper we propose a way of the clauses matrix representation for the resolution algorithm in propositional logic.

## 2. Matrix representation

Let $S = \{S_1, S_2, \dots, S_n\}$is a set of clauses. Let $A = (a_1, a_2, \dots, a_n)$–the alphabet of all letters from S. We assume that A is linearly ordered.

At the initial stage, each clause $S_i$will be represented by ternary set $X_i = \{X_{i1}, \dots, X_{in}\}$ by the rule:

$$X_{ij} = \begin{cases} 1, \text{if} S_\text{i} \text{ includes } \text{a}_\text{j} \text{ without negation} \\ -1, \text{if} S_\text{i} \text{ includes } \text{a}_\text{j} \text{with negation} \\ 0, \text{if} S_\text{i} \text{doesn`t include} \text{a}_\text{j} \end{cases}, \text{ where} j = \overline{1, n}.$$

A set of ternary vectors (sets) corresponding to the initial set S will be obtained as a result of this representation.

The resulting sets should be ordered in some way to simplify the connections search for resolution and avoid increasing of the search space when resolvents are being added.

In this paper, we propose to split sets into *n* classes that contain the amount of zeros corresponds to number of classes. This splitting is equal to splitting by the length of clauses of S. Such classes should be ordered by descending amount of zeros in sets (ascending length of clauses of S). Inside each class sets should be ordered by increasing zero position number.

Each pair of sets $X_l$& $X_j$ is assigned a connection R($l$,$j$)=$\{R_1(l,j), R_2(l,j), \dots, R_n(l,j)\}$ as follows:

$$R_k(l,j) = X_{lk} * X_{jk}, \text{ where } k = \overline{1,n}.$$

## 3. Resolution algorithm on the relations matrix

The proposed algorithm consists of the following procedures: matrix pre-processing, selection of relation for the resolution, the resolution of clauses, adding the resulting vector-resolvent to the relations matrix.

### 3.1 Matrix pre-processing

The optimization procedure on connections matrix involves the removing of disjuncts-tautologies, pure clauses and the disjuncts absorption. Matrix processing is carried out by row and starts from the first row.

1. If the amount of “-1” in connection R(l,j) is greater than one, a disjunct-tautology will be obtained in result of the resolution of pair $C_l$&$C_j$, and such connection should be removed.
2. If $\forall i : i \in \overline{(1,n)}, R_i(l,j) = |R_i(l,l)|$, than disjunct $C_l$ absorbs disjunct$C_j$ and j-s row and column should be removed.
3. If in current row j of matrix R there are no connections for resolution (amount of “-1” is 0), than pure disjunct is obtained and j-s row and column should be removed.

Relation matrix processing continues as long as all rows will be examined. Because absorbing clauses are shorter than absorbed the sets arranging by the amount of zeros allows to execute the absorption procedure in fewer steps.

### 3.2 Selection of resolution connection

An heuristic selection of resolution connections was proposed in [4]. Ternary representation for it will be the following. The connection is selected, if the amount of zeros is different minimally from the amount of zeros of second disjunct and amount of “1” in that connect is maximum. In most cases, this approach allows to avoid high increase of the amount of processed clauses.

A connection selection with help of this heuristic reduces to consecutive selection of resolution connections because of the sets ordering way and building of relations matrix which is proposed in this paper. This approach significantly reduces the number of performed comparisons.

### 3.3 Clauses resolution

The presence of one “0” in the connection sets shows that disjuncts forms a pair contained complementary literals and in this case resolution is possible. Resolution is to obtain from vectors $X_l$& $X_j$ the resolvent – vector, which is defined by the rule:

$$T_k = X_{lk} + X_{jk}, \text{ where } k = \overline{1, n},$$

and “+” is operation which is defined by the table of values:

Table 1 – Values of operation “+”

| + | 0 | 1 | -1 |
|---|---|---|---|
| 0 | 0 | 1 | -1 |
| 1 | 1 | 1 | 0 |
| -1 | -1 | 0 | -1 |

Obtained resolutions are added to the corresponding class of sets without breaking the order inside the class. It allows to reduce the amount of comparisons in the new resolution relation search, and increase the probability of absorption by the new disjunct. If the number of the class of a new set is less than or equal to the number of "parents"  classes, it is necessary to verify the ability of new disjunct to absorb others and to make an absorption if in possible.

It is necessary to define a possibility of added disjunct connection with other disjuncts which set inherits from disjuncts-“parents”. New connections doesn`t appear after resolution.

If at some stage of the algorithm empty resolventderived, then S is satisfiable and the algorithm finishes its work. If at some stage of the algorithm a pair of clauses to resolution can`t be find, S is unsatisfiable.

**Conclusions**

In this paper optimization of inference in propositional logic using the ternary representation clauses is proposed. Heuristics have been redefined at each step of the algorithm: the construction of the relationsmatrix, relationsmatrix preprocessing, operation of absorption and removing of pure clauses, the operation of the resolution. The ternary addition and multiplicationwere described.

On a number of experimental studies have shown that such a representation can reduce the computational complexity by 9%.

**Bibliography**

1. Gomez C.P. Satisfability Solvers/ C.P.Gomez, H.Kautz, A. Sabharawal, B.Selman // Handbook of Knowledge Representation, 2008. – 301 с.

2. Вагин В.Н. Достоверный и правдоподобный вывод в интеллектуальных системах / В.Н. Вагин, Е.Ю. Головина, А.А. Загорянская – М. : ФИЗМАТЛИТ, 2004. – 704 с.

5. Волченко М.В. Алгоритм резолюции в логике высказываний при 0-1-ном представлении дизъюнктов / И.С. Грунский, М.В. Волченко // XII национальная конференция по искусственному интеллекту «КИИ-2010». – М. :Физматлит, 2010. – Т.1. – С.64-71.

6. М.В. Волченко. Оптимизация алгоритма резолюции в логике высказываний при помощи 0,1-ного представления дизъюнктов. Віcникнаціональноготехнічногоуніверситету «ХПІ» серія «Новірішення в сучаснихтехнологіях» № 33. – Х. : НТУ «ХПІ», 2012. С. 55-62.

7. М.В. Волченко. О способе оптимизации матрицы связей для троичного представления дизъюнктов в алгоритме резолюции. Інформаційнітехнології: наука, техніка, технологія, освіта, здоров’я: тезидоповідей XX міжнародноїнауково-практичноїконференції, Ч.IV– Харків, НТУ «ХПІ». С. 9.