Computing Satisfiability Degree of Prepositional Formula

Pragya Verma,Saurabh, R14CS140. Email:pragyaverma345@gmail.com
S Hubham, R14CS177
Gulshan Banu R14CS030

ABSTRACT

Satisfiability dgree is the resource of demonstrating uncertainity. It figures out Weather the scope of prepositional formula be true.Present algorithm calculates the satisfiability degree overlooking the dependency among propositional atoms.we simplify the calculations by using the current algorithm if there is dependencies. Dependency matrix is used to characterize the dependencies. The truth indicator stage of algorithm enumerate the satisfiability degree of a propositional formula. The unproved result shows more efficiency if there is excess dependency among the atoms.

Introduction-

As we know the satisfactory degree adjudge the propositional formula.This decision problem is of central importance in various areas, fields like computer science, circuit design[1],artificial intelligence [2].

Boolean satisfiability problem requires particular and special structure for computing. the two of them are: HORN CLAUSE & COJUNCTIVE NORMAL FORM(CNF)[3]. Horn clause is if it contains more than one positive literals[4]. CNF is if it has conjunction of clauses[5].

Many algorithm comes under CNF which is used to compute satisfiability degree eg: XOBDD algorithm[6], backtracking algorithm[7], proposition matrix search algorithm (PMSA)[8].

The output of PSMA is in matrix form as the dependencies are taken in matrix form in existing algorithm. The proposed dependency propositional matrix search algorithm computes frequency, and chooses the frequency which have maximum value. The advantage over PSMA is DPSMA (dependency propositional search matrix algorithm) as it simplifies the complexity over each iteration and then returns true or false.

DEPENDENCY MATRIX-

1: Let α be a formula which is in Prime Conjunctive Normal Form (PCNF) and q1,q2,...,qn are propositional atoms. let Sα = {q1,q2,...,qn} be a spanning set of formula α. The formula α is given as below.

α = c1 ∧ c2 ∧...∧cm Where c1,c2,...,cm are disjunctive clauses and m ≤ 2n.Here Di = p1∨p2∨...∨pn and pj ∈{qj,¬qj} , 1 ≤ i ≤ n,1 ≤ j ≤ m.

2: As α in PCNF, the satisfiability degree is defined as a function g shown below:

g(α) = (2n−c) /2n

Where α is the number of the s clauses of formula α. In the proposed algorithm, before finding the satisfiability degree of α, proposition matrix is simplified based on the dependencies between the atoms given in the matrix.

3: Let formula α = s1 ∧s2 ∧...∧ sm in CNF and Sα {q1,q2,...,qn} be a spanning set. The proposition matrix of α, Pα have the elements based on following criteria:

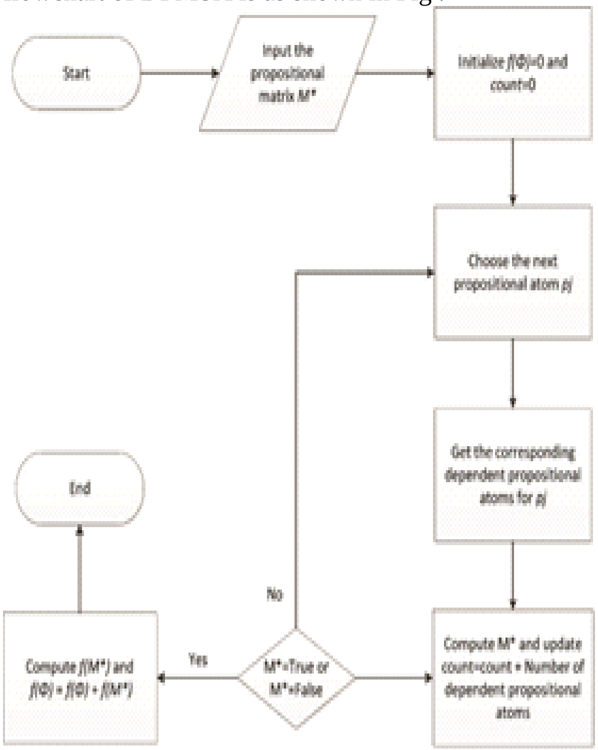
1) P α (i,j) = 1, 1 ≤ i ≤ m, 1 ≤ j ≤ n, denoting qj appears in clause i;
2) P α (i,j) = −1, 1 ≤ i ≤ m, 1 ≤ j ≤ n, denoting qj appears in clause Ci;
3) P α (i,j) = 0, 1 ≤ i ≤ m, 1 ≤ j ≤ n, denoting both qj and ¬qj doesn't appear in clause Ci.
4) Given a formula α = s1 ∧ s2 ∧ ... ∧ sm in conjunctive normal form(CNF). If proposition o has nothing to do with the truth value of formula α, then o is called a free proposition.
   Let W = {o | o is a free proposition} called a free set of α. If proposition o is true, formula α can be simplified into a new formula α +oand a free setW+o.
5) Given formula α = s1∧s2∧...∧sm in CNF and its proposition matrix Pα, the high-frequency proposition q_j on the column can be calculated by using the formula. $X_i = \sum_n^{j=1} |P\alpha\ (j,i)|$ where X_i indicates the frequency of q_j appearing in α.
6) The formula α = s1 ∧s2 ∧...∧sm in conjunctive normal form(CNF) and its proposition matrix . The truth detector finds the sum of row Pα by using the following formula.
   $R_k = \sum_n i^{=1} |\ P\alpha\ (k,i)|$ 1 ≤ k ≤ m If there exists any row with $R_k = 1$, the truth detector chooses the corresponding proposition.
7) Consider a formula α = s1 ∧s2 ∧...∧ sm in CNF and Sα = {q1,q2,...,qn} be a spanning set. The propositional atoms in α may consist of dependencies between them. Proposed algorithm considers the dependency between the propositional atoms in the form of provable equivalence. The dependency matrix of α, Tα have the elements based on following criteria:

1) Tα (i,j) = 1, if qi ⊢ qj, 1 ≤ i ≤ n, 1 ≤ j ≤ n;
2) Tα (i,j) = −1, if qi ⊢¬qj, 1 ≤ i ≤ n, 1 ≤ j ≤ n;
3) Tα (i,j) = 2, if ¬qi ⊢ qj, 1 ≤ i ≤ n, 1 ≤ j ≤ n;
4) Tα (i,j) = −2, if ¬qi ⊢¬qj, 1 ≤ i ≤ n, 1 ≤ j ≤ n;

5) Tα (i,j) = 0, otherwise.

III. PROPOSED ALGORITHM

Dependency matrix is used by DPMSA(dependency propositional matrix serch algoritm)to calculate satisfiability degree of propositional formula .The flowchart of DPMSA is as shown in Fig .



The steps of the algorithm are given below:
Step 1: Input the proposition matrix *Pα*. Initial *count value* =0 and *q*(*α*) = 0, where *count* is used to record the number of propositions that has been processed and the function *q*(*α*) is used to find the satisfiability degree of a given formula *α*. The function *q*(*α*) is defined by the following formula.

$q(\alpha) = q(\alpha j+) + q(\alpha j-)$

Step 2: The next proposition *pj* of α *should be chosen* by using the truth detector. The frequency with only single non zero value should be chosen.
Step 3: to get the equivalent proposition from the matrix add all the corresponding non zero value

. E.g. Consider α = (q1∨¬q2)∧(¬q3∨q4)∧(q2∨q3)∧p4,Sα = {q1,q2,q3,q4}, q4 ⊣⊢¬q1 and chosen proposition p4.
Proposition Matrix,

$$P\alpha = \begin{matrix} 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Dependency Matrix,

$$D\alpha = \begin{matrix} 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{matrix}$$

Here, the simplification process considers both q4 and ¬q1 as they are provably equivalent literals. The simplified expression is ¬q2 ∧(q2 ∨q3).
Step 4: change the value count=count+1 and new atom should be included in the last step.
Step 5: Compute the function *l*, if *N_* = *True* or *N_* =*False*
Step 6: the above steps should be repeated based on the latest high-frequency proposition of the new formula *N_*. This process continues until *N_* can be judged as true or false.

4.EXPERIMENTAL RESULTS

Many experiments where done to verify DPSMA with existing matrix search algorithm. Experiments where done twice for finding the average run time for both DPSMA & PMSA by varying the inputs/dependencies .we will take two tables .In the first table clause=10 & variable=15 this will remain constant for calculation and dependency changes ,and for the second table clause=20 &variable=10.so by performing the experiments we can notice the changes.

Clauses ×Variables ×Dependency	Average time(ms) PMSA	Average time(ms) DPMSA
12×15×0	11.1	10.8
12×15×1	11.1	10.4
12×15×2	11.1	9.6
12×15×3	11.1	8.2
12×15×4	11.1	7.5

Clauses ×Variables ×Dependency	Average time(ms) PMSA	Average time(ms) DPMSA
20×10×0	14.3	14.3
20×10×1	14.3	13.5
20×10×2	14.3	12.6
20×10×3	14.3	11.3
20×10×4	14.3	10.1

5. CONCLUSION

In this paper we have anticipated the staisfiability degree via dependency matrix which reduces the complexity of degree of satisfiablility.we successfully conclude that PMSA is not as effective as DPMSA if there exists more reliance among atoms. in future we can extend the satisfiability degree to check the models,space

REFERENCES

[1] S. Eggersglu and R. Drechsler, “Robust algorithms for high quality test pattern generation using boolean satisfiability,” in *Test Conference (ITC), 2010 IEEE International*, Nov 2010, pp. 1–10.
[2] E. Rich and K. Knight, *Artificial intelligence*. New York: McGraw-Hill,1996.

[3] M. Huth and M. Ryan, *Logic in Computer Science: Modelling and Reasoning About Systems*. New York, NY, USA: Cambridge University Press, 2004.
[4] A. Horn, "On sentences which are true of direct unions of algebras," *J. Symb. Log.*, vol. 16, no. 1, pp. 14–21, 1951. [Online]. Available: http://dx.doi.org/10.2307/2268661
[5] E. Mendelson, Introduction to Mathematical Logic, 4th ed. London: Chapman and Hall, 1997.
[6] G. Luo, C. Yin, and P. Hu, "An algorithm for calculating the satisfiability degree," in Fuzzy Systems and Knowledge Discovery, 2009. FSKD '09. Sixth International Conference on, vol. 7, Aug 2009, pp. 322–326.
[7] C. Yin, G. Luo, and P. Hu, "Backtracking search algorithm for satisfiability degree calculation," in Fuzzy Systems and Knowledge Discovery, 2009. FSKD '09. Sixth International Conference on, vol. 2, Aug 2009, pp. 3–7.
[8] J. Luo and G. Luo, "Proposition matrix search algorithm for satisfiability degree computation," in Cognitive Informatics (ICCI), 2010 9th IEEE International Conference on, July 2010, pp. 974–977.
[9] Shreekant Jere REVA ITM, Bangalore "Computation of Satisfiability Degree of a Propositional Formula using Dependency Matrix".