Satisfiability Degree Analysis for Transition System*

Yang Zhao and Guiming Luo

School of Software, Tsinghua University
Beijing 100084, China
yang-zhao08@mails.tsinghua.edu.cn, gluo@tsinghua.edu.cn

Abstract. Classical model checking is not capable of solving the situation of uncertain systems or fault-tolerant systems which occur in the real world commonly. The technique of satisfiability degree (SD) for model checking is a efficient way to solve this problem. Finite paths transition system (FPTS) for calculating satisfiability degree of a LTL logic formula is given. Then, a more general situation about discrete-time Markov chains (DTMCs) is discussed. Then a case named leader election shows the practicability of satisfiability degree for transition system, which cannot be solved by classical model checking.

Keywords: satisfiability degree, transition system, LTL.

1 Introduction

Classical logic is related certainty, i.e., any proposition is either true or false, with no other intermediate states. However, the real world does not always reflect the classical logic since there are many cases with neither absolute true nor absolute false. So many studies have used non-classical logic, such as modal logic, intuition logic, fuzzy logic, and probabilistic logic. Paper [1] analyzes the satisfiability of a proposition. The calculus of satisfiability degree is proposed, which expresses uncertainty, extends the concepts of satisfactory and contradictory propositions in propositional logic and describes how satisfactory is based on the proportion of interpretations that make the proposition true.

In this paper we want to provide a new sight of model checking to deal with the uncertain situation. In the real world, may sometimes the system can not satisfy the property totally, but it can support partially. Then classical model checking methods have difficulties to describe these cases. The conception of Satisfiability Degree is introduced into model checking, to calculate the satisfiability degree of a logic formula to express the level about a transition system satisfy the formula.

The paper is organized as follows: Section 2 introduces the definition of satisfiability degree in Propositional Logic. Section 3 discusses the satisfiability degree on transition systems, extending to the system FPTS and DTMCs. Section 4 gives a simple example to show the application of satisfiability degree on transition systems, leader election. The conclusions are given in the last section.

* This work is supported by the Funds NSFC 60973049 and 60635020.

F.L. Wang et al. (Eds.): AICI 2010, Part I, LNAI 6319, pp. 25–32, 2010.

2 Satisfiability Degree

Let P be a propositional formula set and Ω the global field for interpreting P, thus any formula $\varphi \in P$ and $\omega \in \Omega, \varphi(\omega) \in \{0,1\}$. If $\varphi \in P$, define a subset $\Omega\varphi \subset \Omega$ such that:

$$\Omega\varphi = \{\omega \mid \varphi(\omega) = 1, \omega \in \Omega\} \tag{1}$$

Definition 1 (Satisfiability Degree in propositional logic)
Given a propositional formula set **P** and the global interpretation field Ω, the subset is defined as above. Then, function $f: P \to [0,1]$ is called the satisfiability degree on Ω, if any formula $\varphi \in \mathbf{P}$:

$$f(\varphi) = d(\Omega\varphi)/\, d(\Omega) \tag{2}$$

where d(X) is denote as the cardinality of set X.

The most intuitive method for computing the satisfiability degree is enumeration of all the assignments using simulation as a truth table. However the complexity of this method is exponential. Three methods are presented to reduce the unnecessary costs [1-3]. One is base on the satisfiability degree properties, while another is constructed as an Extended Ordered Binary Decision Diagram (XOBDD) [2, 3]. The backtracking search algorithm for SAT is used and optimized modification also can help speed up the calculation.

3 Satisfiability Degree for Transition System

As we know, the concept of satisfiability degree on propositional logic is come up for the satisfiable properties of Propositional Logic formulas. Similarly, in field of model checking the basic idea is that whether a transition system S is a model of a logic formula F must be checked, that is, we want to know whether S can make F satisfiable.

Definition 2 (Finite paths transition system (FPTS))
Finite paths transition system is a seven-triple (S, δ, I, F, AP, L) where

- S: is a finite set of states;
- $\delta \subseteq S \to S$ is a transition relation function;
- $I \subseteq S$ is a set of initial states, which means for any s in I, there is no state t in S so that $\delta(t) = s$;
- $F \subseteq S$ is a set of final states, which means for any s in F, there exist no state t in S so that $\delta(s,) = t$;
- AP is a set of atomic propositions, and L: $S \to 2^{AP}$ is a labeling function.
- We write $\delta(s_i,) = s_j$, if s_i can transfer to s_j directly, $\delta^k(s_i) = s_j$ is written to show s_i can transfer to s_j by k steps; so the last conditions is given as follows,
- For any state s in S, there is no k exist so that $\delta^k(s) = s$, where $k \in N$;

Definition 3 (Path)
Let D be an FPTS, a path ρ in D is an sequence $(s_0, s_1, s_2 \ldots, s_n)$, since s_0 is one in I ,and s_n in F, and for any i in N, there exist $\delta(s_i) = s_{i+1}$

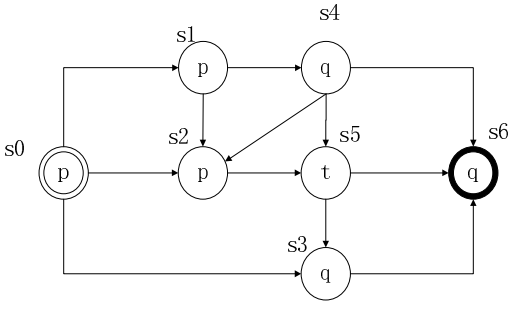


Fig. 1. A simple FPTS example

Figure 1 show an example about FPTS , AP={p, q, t}, and L is given L(s0) = L(s1) = L(s2)={p}, L(s3)=L(s4)=L(s6)={q}, L(s5)={t}. Many paths such as {s0, s2, s5, s6} and {s0, s1, s4 s5, s6} can be found, but the amount of the paths are finite, and for each path it has finite states.

On FPTS we can give the satisfiability degree definition of a LTL formula easily; the definition of linear-time temporal logic (LTL) can be found in [5]. Then for a LTL formula, we may give the definition of Satisfiability Degree on FPTS, to show the level FPTS satisfy a LTL formula.

Definition 4 (SD on FPTS)
Given an FPTS D=(S, Act, δ, I, F, AP, L), P is a LTL formula set, then the function $f_D: P \to [0,1]$ called Satisfiability Degree on D, if for any formula $\varphi \in P$:

$$f_D(\varphi) = \frac{NP(D|\varphi)}{NP(D)}, \tag{3}$$

Where NP(D) means the number of paths in D, and $NP(D|\varphi)$ is denoted as the number of paths in D which satisfy the formulaφ.

Next, the algorithm is provided to compute the SD on FPTS. we may modify the classical algorithm DFG (Deep First Search) [6] [8] to make it work. SD= (m, n) is denoted where both m and n are integers. We calculate the satisfiability degree by SD using m minus n so from the algorithm we will get SD;

Algorithm 1. Get satisfiability degree of a LTL formula on a FPTS;

```
1:  m := 0;
2:  n := 0;
3:  path :=null; // path is a stack, also represent a path
4:  for each s in S
5:    S_s= {t|δ(s,act) = t, if there exist act in Act}
6:  for each s in I
7:     {
8:     SD sd := SD-Visit(0,s);
9:     m := m + sd.m;
10:    n := n + sd.n;
11:    }
12:  return (m, n)
```

Algorithm 2. SD-Visit (s, t)

```
1:  SD sd(m, n) :=(0, 0);
2:  Push (path, t);
3:  if t is in F {
4:     sd.m := sd.m + CheckLTL(path, φ);
5:     sd.n := sd.n + 1; }
6:  else {
7:      for each u in St {
8:         SD sd2(m, n) = SD-Visit(t, u);
9:         sd.m := sd.m + sd2.m;
10:         sd.n := sd.n + sd2.n; } }
11:    Pop (path, t);
12:    return sd;
```

Algorithm 3. CheckLTL(path, φ)

```
1: if path have not satisfied φ
2:    return 0;
3: else return 1;
```

It is trivial that n is the total number of all paths in this FPTS and m represents the number of paths satisfy the formula. So from the definition, we can easily get the SD by m/n. The worst situation may cost no more than $O(2^n)$, for n represents the number of states in S. And the best situation may cost O(n), when only one path exists.

Definition 5(DTMCs)
A (labeled) discrete-time Markov chain (DTMC) is a triple D=(S, P, L) where:

- *S* is a finite set of states;
- P: $S \times S \rightarrow [0,1]$ is a stochastic matrix;
- L: $S \rightarrow 2^{AP}$ is a labeling function which assigns to each state $s \in S$ the set L(s) of atomic propositions that are valid in S

Definition 6(Paths)
Let D=(S, P, L) be a DTMC. An infinite path ρ in D is an infinite sequence $(s_0, s_1, s_2 \ldots, s_n)$ of states such that for all i>0, P(si,si+1)> 0.

A finite path σ is a finite prefix of an infinite path.

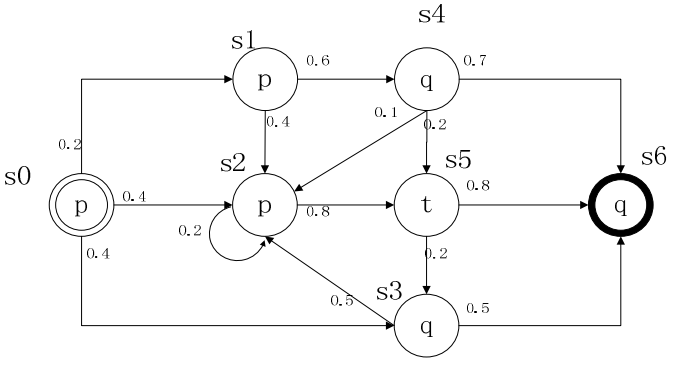


Fig. 2. A example about DTMCs

A DTMCs is given in Figure 2 which is modified from the one in figure1. The difference is that The state s2 has a self-circle and s2 can transfer to itself through s5 and s3. So obviously it can have infinite paths. And every paths has a weight.

We denote the Ω is the set of all paths in a DTMCs D, and φ is a LTL formula, let satisfied set Ωφ={ρ | ρ satisfy φ, ρ in D} a subset of Ω. And characteristic function is denoted like this:

$$X\varphi(\rho)=\begin{cases} 1 & (\text{if } \rho \in \Omega\varphi) \\ 0 & (\text{if } \rho \in \Omega \setminus \Omega\varphi) \end{cases} \tag{4}$$

Then the definition of satisfiability degree is denote as below:

Definition 7 (**SD on DTMCs**)
Let D=(S, P, L) be a DTMC, $\sum$ is a LTL formula set, then the function $f_D: \sum \to [0,1]$ called satisfiability degree on D, if for any formula $\varphi \in \sum$:

We have

$$f_D(\varphi) = \sum_{\rho \in \Omega} W(\rho) X\varphi(\rho) \tag{5}$$

We also can write $f_D(\varphi)$ as $f(\varphi|D)$.

The form of SD definition for DTMCs is very different from FPTS for the probability and infinite situation considered, so in order to compute SD we must transfer the infinite situation to finite situation (mainly caused by self-circles).

The difficulty is how to deal with self-circles. As we know LTL has some operator as X, F, G, U, W, R. For such as X, we can just check the next state to get the result, another word, and finite step must be finished, but the others we must check more states, maybe infinitely.

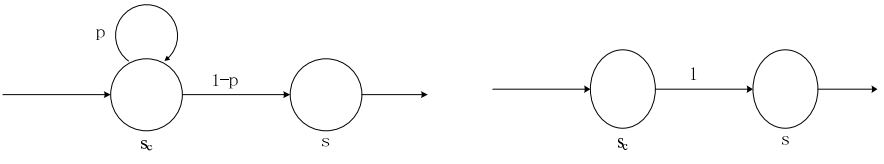


Fig. 3. Simplification of self-circles

Considering when just arriving a state s_c owning a self-circle and as it happens we just right check the formula Fφ, whereφ is a formula. If $L(s_c)$ can make φ right, then the path

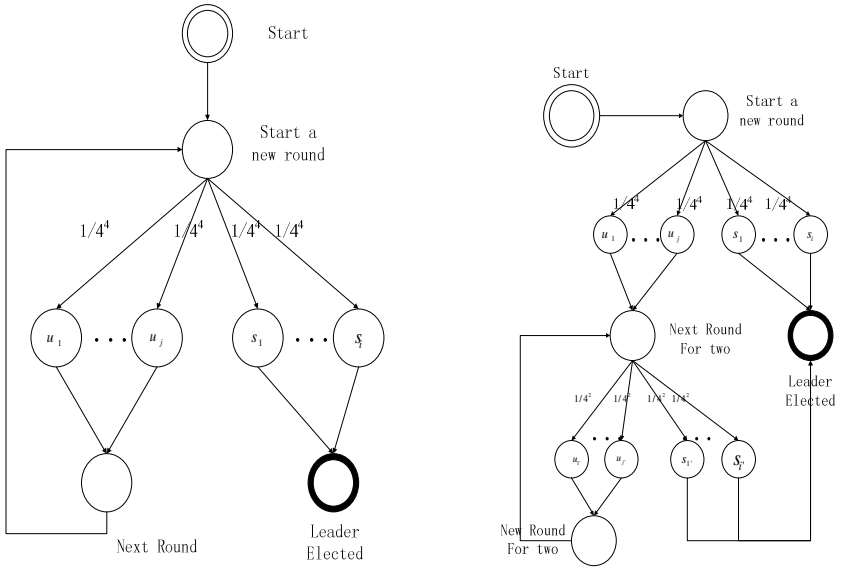
$$* s_c s *, * s_c s_c s *, \quad * \underbrace{s_c \ldots s_c}_{i} s *$$

all satisfy the formula φ where i can be any positive integer and * represents the other states before sc and after s. Vice versa. So if wrong, we can stop checking the next sc, and back-track to the previous one. And if right, we just calculate the sum $\sum_{i=0}^{\infty} p^i(1-p)=1$, and continue to check the next one. So the situation of figure 3(a) can be changed to the figure3 (b) as below.

4 Applications

In this chapter a case called Leader Election [7] is introduced to show how our satisfiability degree on transition system works. Without loss of generality, we consider four electors select one leader from their selves. The one who will be the leader must have more than the half, that is, must be selected by three people (including himself). Every person is a candidate while a elector, who has a unique id selected from {1,2,3,4} We have two ways: First, everyone can vote one (including himself) randomly, then check the result, if someone has three or more supporters, then he win, otherwise we enter the next round and do it again, until someone become the leader. Second method, we do the same thing on the first round, but if no one is selected, we enter the second round, but the rule is changed, we choose only two candidates who have the most supporters (if there exist equal votes, select the one has a bigger id for convenience), then vote again, we check if still none have three, vote the two one again. All vote process is random. Since every round may cost resource, so we cannot make the election round and round, never stop. So we want to know if we want have the leader only in two rounds, then which way is better.

To solve the problem, we may model the two ways as below:



(a) The Model of first way (b) The Model of second way

Fig. 4. The models of leader election

Let $D_1=(S_1,P_1,L_1)$ be a DTMCs, described as Figure 4 (a), $D_2=(S_2,P_2,L_2)$ be a DTMCs, described as Figure 4 (b), where $u_1,..,u_j$ are the states representing the voting situation about unable to get a leader, $s_1,..,s_i$ means getting a leader successfully. Since the vote is random, the probability of arriving at u or s is $1/4^4$, the same way for $u_{1'},\ldots,u_{j'}$, $s_{1'}\ldots s_{i'}$ with $1/4^2$. We will simplify them by calculating the total probability

of leader elected on the first round and second round, and combine the states .

Leader must have at least three tickets, so on the first round, the probability of leader elected can calculate as

$$\frac{4 \times C_4^4 + 4 \times C_4^3 \times C_3^1}{4^4} = \frac{13}{64}$$

while none elected on the first round

$$1 - \frac{13}{64} = \frac{51}{64}$$

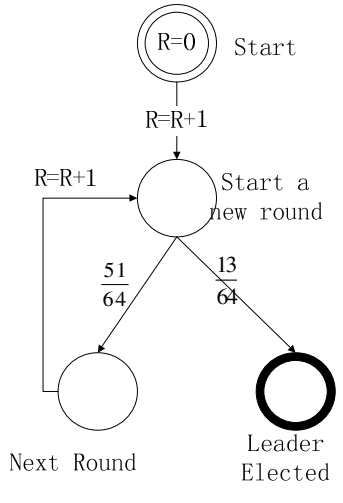
On the second round, while the second way only has two candidates left, the probability can calculate as

$$\frac{2 \times C_4^4 + 2 \times C_4^3}{2^4} = \frac{5}{8}$$

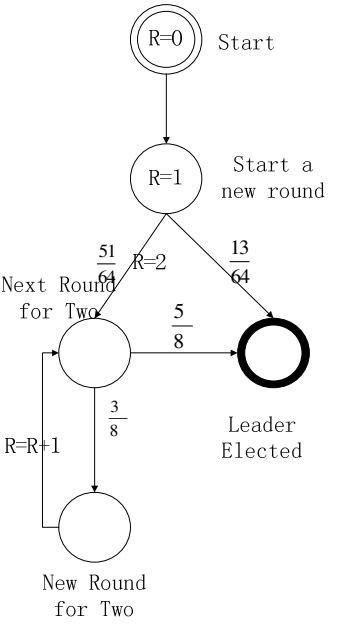
while none elected on the second round

$$1 - \frac{5}{8} = \frac{3}{8}$$

So we simplify both D1 and D2 as below:



(a) The Simplification of Fig 4(a)



(b) The Simplification of Fig 4(b)

Fig. 5. The Simplification of Fig 4 by combining similar states

Then a LTL formula can be used describe the property we concern: we want have the leader only in two rounds. The formula is written as follows:

$$\varphi = \mathrm{F}\,(\mathrm{Leader\_Elected} \wedge \mathrm{R} \leq 2) \tag{6}$$

Now the satisfiability degree of φ on D1 and D2 can be computed

$$\mathrm{f}(\varphi|\mathrm{D}_1) \cong 0.371,$$

$$\mathrm{f}(\varphi|\mathrm{D}_2) \cong 0.701.$$

Obviously, $\mathrm{f}(\varphi|\mathrm{D}_2) > f(\varphi|\mathrm{D}_2)$, that is, though both D1 and D2 have the ability to satisfy φ true, but D2 is more stronger than D1, so we want to have the leader only in two rounds, the second way is better than the first, which is told by their satisfiability degrees.

5 Conclusions

In this paper, the conception of satisfiability degree is introduced into model checking, which is extend from based on propositional Logic to the higher-order logic mainly LTL through the transition system. We not only give the definition of satisfiability degree on transition system but the algorithm of computation on the systems including FPTS and DTMCs. This algorithm modifies the DFS algorithm so that satisfiability degree can be calculated easily. The application of satisfiability degree on transition system is also shown to check which system can satisfy a property better by the example leader election. We show the way how model checking can do quantitative analysis and fault-tolerant analysis for sometimes we must compromise to the real world if we cannot have the certain result, and then the satisfiability degree is a good choice.

References

[1] Luo, G.M., Yin, C.Y., Hu, P.: An algorithm for calculating the satisfiability degree. In: Proceedings of the 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2009), vol. 7, pp. 322–326 (2009)

[2] Yin, C.Y., Luo, G.M., Hu, P.: Backtracking search algorithm for satisfiability degree calculation. In: Proceedings of the 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2009), vol. 2, pp. 3–7 (2009)

[3] Hu, P., Luo, G.M., Yin, C.Y.: Computation of satisfiability degree based on CNF. In: Proceedings of the 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2009), vol. 6, pp. 142–146 (2009)

[4] Clarke, E.M., Orna, G., Peled, D.A.: Model Checking, pp. 13–24. The MIT Press, Cambridge (1999)

[5] Huth, M.l., Ryan, M.: Logic in Computer Science Modeling and Reasoning about Systems, 2nd edn., pp. 172–186 (2005)

[6] Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Intruduction to algorithm, 2nd edn., pp. 540–549 (2002)

[7] Itai, A., Rodeh, M.: Symmetry Breaking in Distributed Networks. Information and Computation 88(1), 60–87 (1990)

[8] Hong, I., Sohn, S.H., Lee, J.K., et al.: DFS algorithm with ranking based on genetic algorithm in UHF portable system. In: 2009 9th International Symposium on Communications and Information Technology, ISCIT 2009, pp. 454–459 (2009)