Project Documentation

Course: Logic For Computer Science

Instructor: Sujata Ghosh

Topic: **Modal Satisfiability problems**

Name: Basudeb Roy

Student Id: JRF Scholar

April 17, 2024

Contents

1 Introduction

Logic is about expressing and proving constraints on mathematical models. There exists various logic to do that. From the computer scientist's perspective, it is obvious questions come into the mind about computability and complexity. In this presentation, we investigate the computability of satisfiability problems. Here we will prove that the Modal Satisfiability problem belongs to PSPACE.

The remainder of this paper is structured as follows. In Sect. 2 we recall some definitions. In Sect. 3 we define graph predicates and provide a translation to FOL; in Sect. 4 we define the reverse translation. Finally, in Sect. 5 we summarize the results and discuss related work, variations, and extensions.

2 Basic Definition

2.1 Modal Logic:-

So Before we define modal logic let us first understand the the meaning of modality. So, a modality is just a word or phrase that can be applied to a given statement S to create a new statement that asserts. There exist different types of modality. For example, temporal, epistemic, preference, deontic, dynamic, metalogic, etc. Now, we are ready to define modal logic. So, a Kripke modal logic constitutes:

- A set W of states or worlds(each one specifying truth values for all propositional variables).

- A relation on the set of states(specifying the 'relevant situations'). Therefore, we viewed this relation R $\subseteq$ W $\times$ W.

- So, a frame $\mathcal{F}$ is a pair of the W and R i.e., $\mathcal{F} = (W, R)$.

- A model is a pair $\mathcal{M} = (\mathcal{F}, V)$. Where V is a valuation function such that $V : W \to 2^P$.

Now, given a model, $\mathcal{M}$ and a world w in $\mathcal{M}$ is called a 'pointed model'. Now, in Kripke modal logic any propositional variable is a formula. If P and Q are formulas, then $\neg P, P \wedge Q, P \vee Q, P \to Q$, $\Box P$, $\diamond P$ are also formulas. For example, in the following figure 1 the set of world W = $\{w_1, w_2, w_3, w_4, w_5\}$. The valuation functions are V(P) = $\{w_1, w_5\}$, V(Q) = $\{w_2, w_3, w_5\}$ and V(R) = $\{w_4\}$. P, Q, and R are the propositional variables.

2.2 Quantified Boolean Formula:-

The quantified Boolean formula problem (QBF) is a generalization of the Boolean satisfiability problem in which both existential quantifiers and universal quantifiers can be applied to each variable. Put another way, it asks whether a

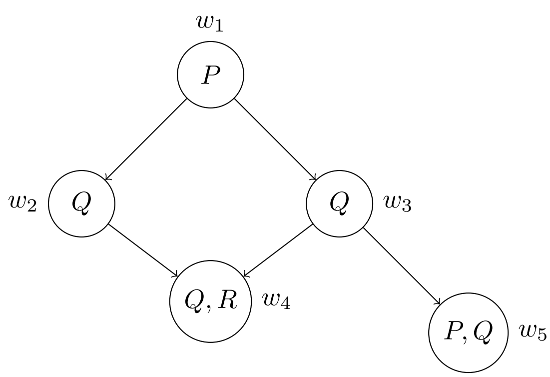

Figure 1: Example of modal logic

quantified sentential form over a set of Boolean variables is true or false. For example, the following is an instance of QBF:

$$\forall x \; \exists y \; \exists z \; ((x \vee z) \wedge y)$$

3 PSPACE:-

PSPACE, the class of problems solvable by a deterministic Turing machine using only polynomial space, is the complexity class of most relevance to the basic modal language. Intuitively, PSPACE-complete problems are the hardest problems in this category. Here are some problems that are known to be PSPACE-complete:

- Given a regular expression, does it match all possible strings, or is there a string it doesn't match?

- Given a formula with no free variables, such as " $\exists x_1 \forall x_2 \exists x_3 \forall x_4 : (x_1 \vee \neg x_3 \vee x_4) \wedge (\neg x_2 \vee x_3 \vee \neg x_4)$", check whether it is true.

In this presentation, we will show that modal satisfiability problems belong to PSPACE. We will show that modal logic does not have the polysize model property.

4 Forcing binary trees in Modal Logic:-

Let ϕ^β be a satisfiable formula. For any natural number m, we are going to devise a satisfiable formula $\phi^\beta(m)$ with the following properties:

- the size of $\phi^\beta(m)$ is polynomial (indeed, quadratic) in m,but

- when ϕ^β is satisfied in any model $\mathcal{M}$ at a node w_0, then the submodel of $\mathcal{M}$ generated by w_0 contains an isomorphic copy of the binary tree of depth m.

As we know that the binary branching tree of depth m contains 2^m nodes, the size of the smallest satisfying model of $\phi^\beta(m)$ is exponential in $|\phi^\beta(m)|$. Thus we will have shown that small formulas can force the existence of large models.

We will define these formulas by mimicking truth tables. For any natural number m, $\phi^\beta(m)$ will be constructed out of the following variables: $q_1, \cdots, q_m$, and $p_1, \cdots, p_m$. The q_i's play a supporting role. They will be used to mark the level (or depth) in the model; that is, they will mark the number of upward steps that need to be taken to reach the satisfying node. But any satisfying model for $\phi^\beta(m)$ will give rise to a full truth table for $p_1, \cdots, p_m$: every possible combination of truth values for $p_1, \cdots, p_m$ will be realized at some node, and hence any model for $\phi^\beta(m)$ must contain at least 2^m nodes. Let us first define two macros to carry out the above idea.

1. **B_i:-** B_i is defined as follows:

$$B_i := q_i \rightarrow (\diamond(q_{i+1} \wedge p_{i+1}) \wedge \diamond(q_{i+1} \wedge \neg p_{i+1})) \tag{1}$$

   Given that we are going to use the q_i's to mark the levels, the effect of B_i should be clear: it will force a branching to occur at level i, set the value of p_{i+1} to true at one successor at level i +1, and set p_{i+1} to false at another.

2. **S(p_i, $\neg p_{i+1}$):=** S(p_i, $\neg p_{i+1}$) is defined as

$$S(p_i, \neg p_{i+1}) := (p_i \rightarrow \Box p_i) \wedge (\neg p_i \rightarrow \neg\Box p_i) \tag{2}$$

   This formula sends the truth values assigned to p_i and its negation one level down. The idea is that once B_i has forced a branching in the model by creating a p_{i+1} and a $\neg p_{i+1}$ successor, S(p_{i+1}, $\neg p_{i+1}$) ensures that these newly set truth values are sent further down the tree; ultimately we want them to reach the leaves.

Now, we will define the $\phi^\beta(m)$, which is the conjunction of the formula listed below:-

1. q_0
2. $\Box^m(q_i \wedge_{i \neq j} q_j)$ (0 ≥ i ≥ m)
3. $B_0 \wedge \Box B_1 \quad \wedge \Box^2 B_2 \quad \wedge \Box^3 B_3 \quad \wedge \cdots \wedge \Box^{m-1} B_{m-1}$
4. $$\begin{aligned} \Box S(p_1,\neg p_1) \wedge \Box^2 S(p_1,\neg p_1) \wedge \Box^3 S(p_1,\neg p_1) \wedge \cdots \wedge \Box^{m-1} S(p_1,\neg p_1) \\ \wedge \Box^2 S(p_2,\neg p_2) \wedge \Box^3 S(p_2,\neg p_2) \wedge \cdots \wedge \Box^{m-1} S(p_2,\neg p_2) \\ \wedge \Box^3 S(p_3,\neg p_3) \wedge \cdots \wedge \Box^{m-1} S(p_3,\neg p_3) \\ \vdots \\ \wedge \Box^{m-1} S(p_{m-1},\neg p_{m-1}) \end{aligned}$$

The first conjunct q_0, ensures that any node that satisfies $\phi^\beta(m)$ is marked as having level 0. The effect of (2) is to ensure that no two distinct level marking atoms q_i and q_j can be true at the same node (at least, this will be the case out to level m, which is all we care Thus our level markers are beginning to work as promised. about). To see this, recall that $\Box^{(m)}\phi$ is shorthand for $\phi \wedge \Box\phi \wedge \Box^2\phi \wedge \cdots \wedge \Box^m\phi$. Thus our level markers are beginning to work as promised. Because of the prefixed blocks of $\Box$ modalities, the B_i macros in (iii) force m successive levels of branching; and each such branching 'splits' the truth value of one of the p_i's. Then, again because of the prefixed 2 modalities, (iv) uses the $S(p_i, \neg p_i)$ macro to send each of these newly split truth values all down to the m-th level. In short, (iii) creates branching, and (iv) preserves it. So, it is clear that any satisfying model for $\phi^\beta(m)$ must contain a submodel that is isomorphic to the binary branching tree of depth m. It follows that any model of $\phi^\beta(m)$ must contain at least 2^m nodes, as we claimed. Hence, the following theorem comes into this place. Similarly, each row in (iv) gains an extra conjunct (as does the next empty row) thus we gain a new column containing m formulas. The biggest change occurs in (ii). If you write (ii) out in full, you will see that it gains an extra row, and an extra column, and an extra atomic symbol in each embedded conjunct, and this means that —$\phi^\beta(m)$— increases by O(m^2log(m)) (that is, slightly faster than quadratically). This is negligible compared with the explosion in the size of the smallest satisfying model: this doubles in size every time we increase m by one.

Theorem 1 *Modal logic lacks the polysize model property.*

5 PSPACE algorithm for Modal Logic:-

We will now define a PSPACE algorithm called Witness whose successful termination guarantees the modal logic satisfiability of the input. Since we have just seen that there are satisfiable formulas $\phi^\beta(m)$ whose smallest satisfying model contains 2^m nodes.

Definition 1 *A set of formulas Σ is said to be **closed** if it is closed under subformulas and single negations.*

Definition 2 *If Γ is a set of formulas, then Cl(Γ), the closure of Γ, is the smallest closed set of formulas containing Γ. Note that if Γ is finite then so is Cl(Γ).*

Definition 3 *Let Σ be a subformula closed set of formulas. A Hintikka set H over Σ is a maximal subset of Σ that satisfies the following conditions:*

1. $\bot \notin H$.
2. *If $\neg\phi \in \Sigma$, then $\neg\phi \in H$ iff $\phi \notin H$.*
3. *If $\phi \wedge \psi \in \Sigma$, then $\phi \wedge \psi \in \Sigma$ iff $\phi \in H$ and $\psi \in H$.*

If a Hintikka set is satisfiable we call it an atom. Witness will take two finite sets of formulas H and Σ as input, and determine whether or not H is an atom over Σ. next, we will define the demands.

Definition 4 *Suppose H is a Hintikka set over Σ, and $\diamond\psi \in H$. Then the demand that $\diamond\psi$ creates in H is*

$$\{\psi\} \cup \{\sim \theta | \neg \diamond \theta \in H\}$$

The demand is denoted by Dem(H, $\diamond\psi$). Now, we are ready to give the algorithm to check whether the witness is satisfiable or not.

Definition 5 *Suppose H and Σ are finite sets of formulas such that H is a Hintikka set over Σ. Then $\mathcal{H} \subseteq P(\Sigma)$ is a witness set generated by H on Σ if H $\in \mathcal{H}$ and*

1. *if $I \in \mathcal{H}$, then for each $\diamond\psi \in I$, there is a $J \in I_{\diamond\psi}$ such that $J \in \mathcal{H}$*
2. *if $J \in \mathcal{H}$ and $J \neq H$ then for some n¿ 0 there are $I^0,\ldots ,I^n \in \mathcal{H}$ such that $H = I^0$, $J = I^n$, and for each 0 $\geq$ i ¡ n there is some formula $\diamond\psi \in I_i$ such that $I^{i+1} \in I^i_\diamond\psi$.*

Algorithm 1 Witness(H,Σ)

Require: H, Σ
Ensure: boolean
  if H is a Hintikka set over Σ and for each subformula $\diamond\psi \in$ H there is a set of formula I $\in H_{\diamond\psi}$ such that witness(I, Cl(Dem(H, $\diamond\psi$))) **then**
    return true
  else if then
    return false

---

6 Conclusion

We have presented that the modal satisfiability problem is PSPACE-complete.