Complexity of Modal Logic(Part 2)

Tang Wen

May 21, 2020

Contents

1.*K-WORLD* Algorithm

K-WORLD Algorithm

To show that the **K**-satisfiablity problem is in PSPACE, we give a PSPACE algorithm for deciding whether a formula is satisfiable by a **K**-model.

First we define the algorithm *K-WORLD*$(\Sigma_1, \Sigma_2, \Sigma_3, \Sigma_4)$ with the four inputs are all finite sets of formulas.

The value of *K-WORLD*$(\Sigma_1, \Sigma_2, \Sigma_3, \Sigma_4)$ is **true** iff there is a **K**-model $\mathcal{M}$ and $w \in W_{\mathcal{M}}$ such that

$$\mathcal{M}, w \models \bigwedge_{\phi \in \Sigma_1} \phi \wedge \bigwedge_{\phi \in \Sigma_2} \neg\phi \wedge \bigwedge_{\phi \in \Sigma_3} \Box\phi \wedge \bigwedge_{\phi \in \Sigma_4} \neg\Box\phi$$

K-WORLD Algorithm

procedure $K\text{-}WORLD(\Sigma_1, \Sigma_2, \Sigma_3, \Sigma_4)$:
begin
if $\Sigma_1 \cup \Sigma_2 \not\subseteq \mathbf{P}$ **then**
begin
choose $\phi \in \Sigma_1 \cup \Sigma_2$ and $\phi \notin \mathbf{P}$
if $\phi = \neg\psi$ and $\phi \in \Sigma_1$, **then return** $K\text{-}WORLD(\Sigma_1 - \{\phi\}, \Sigma_2 \cup \{\psi\}, \Sigma_3, \Sigma_4)$
if $\phi = \neg\psi$ and $\phi \in \Sigma_2$, **then return** $K\text{-}WORLD(\Sigma_1 \cup \{\psi\}, \Sigma_2 - \{\phi\}, \Sigma_3, \Sigma_4)$
if $\phi = \psi_1 \wedge \psi_2$ and $\phi \in \Sigma_1$, **then return** $K\text{-}WORLD((\Sigma_1 \cup \{\psi_1, \psi_2\}) - \{\phi\}, \Sigma_2, \Sigma_3, \Sigma_4)$
if $\phi = \psi_1 \wedge \psi_2$ and $\phi \in \Sigma_2$, **then return** $K\text{-}WORLD(\Sigma_1, (\Sigma_2 \cup \{\psi_1\}) - \{\phi\}, \Sigma_3, \Sigma_4) \vee K\text{-}WORLD(\Sigma_1, (\Sigma_2 \cup \{\psi_2\}) - \{\phi\}, \Sigma_3, \Sigma_4)$
if $\phi = \Box\psi$ and $\phi \in \Sigma_1$, **then return** $K\text{-}WORLD(\Sigma_1 - \{\phi\}, \Sigma_2, \Sigma_3 \cup \{\psi\}, \Sigma_4)$
if $\phi = \Box\psi$ and $\phi \in \Sigma_2$, **then return** $K\text{-}WORLD(\Sigma_1, \Sigma_2 - \{\phi\}, \Sigma_3, \Sigma_4 \cup \{\psi\})$
end
if $\Sigma_1 \cup \Sigma_2 \subseteq \mathbf{P}$ **then**
begin
if $\Sigma_1 \cap \Sigma_2 \neq \emptyset$ **then return false**
if $\Sigma_1 \cap \Sigma_2 = \emptyset$ **then return** $\bigwedge_{\phi \in \Sigma_4} K\text{-}WORLD(\Sigma_3, \{\phi\}, \emptyset, \emptyset)$
end
end
(note that the conjunction over emptysets is to return **true**)

K-WORLD Algorithm

First we show that *K-WORLD* is indeed correct

Proof.

We define $degn(\phi)$ to be the connectives$(\neg, \wedge, \Box)$ in ϕ. $degn(\Sigma) = \sum_{\phi \in \Sigma} degn(\phi)$, then we let $n = degn(\Sigma_1) + degn(\Sigma_2) + degn(\Sigma_3) + degn(\Sigma_4)$, we show the correctness by induction on n:

$n = 0$: In this case all the sets only contain propositional letters. And we have that the desired formula is satisfiable if and only if $\Sigma_1 \cap \Sigma_2 = \emptyset$ and $\Sigma_3 \cap \Sigma_4 = \emptyset$(from right to left, consider the $\mathcal{M}$ where $W = \{w, v\}$,$R = \{\langle w, v\rangle\}$ and V such that $V(w) = \Sigma_1$, $V(v) = \Sigma_3$. The converse direction can be easily shown by proof by contradiction). And according to the algorithm, we have that $K\text{-}WORLD(\Sigma_1, \Sigma_2, \Sigma_3, \Sigma_4)$ returns **true** iff $\Sigma_1 \cap \Sigma_2 = \emptyset$ and $\Sigma_3 \cap \Sigma_4 = \emptyset$

K-WORLD Algorithm

Proof.

$n > 0$: If $\Sigma_1 \cup \Sigma_1 \not\subseteq \mathbf{P}$, we only show the case where $\phi = \neg\psi$ and $\phi \in \Sigma_1$ and the rest are analogue. According to algorithm, the recursion call lead to *K-WORLD*$(\Sigma_1 - \{\phi\}, \Sigma_2 \cup \{\psi\}, \Sigma_3, \Sigma_4)$, by definition we can have that $\deg(\Sigma_1 - \{\phi\}) + \deg(\Sigma_2 \cup \{\psi\}) + \deg(\Sigma_3) + \deg(\Sigma_4) = n - 1$, by induction hypothesis, we have that *K-WORLD*$(\Sigma_1 - \{\phi\}, \Sigma_2 \cup \{\psi\}, \Sigma_3, \Sigma_4)$ return **true** iff there is a **K**-model $\mathcal{M}$ and $w \in W_{\mathcal{M}}$ such that

$$\mathcal{M}, w \models \bigwedge_{\phi \in \Sigma_1 - \{\phi\}} \phi \wedge \bigwedge_{\phi \in \Sigma_2 \cup \{\psi\}} \neg\phi \wedge \bigwedge_{\phi \in \Sigma_3} \Box\phi \wedge \bigwedge_{\phi \in \Sigma_4} \neg\Box\phi$$

Hence we can have that the proposition holds in this situation. □

K-WORLD Algorithm

Now we can define a algorithm to test whether a formula ϕ is **K**-satisfiable:

begin
  read ϕ
  $v \leftarrow K\text{-}WORLD(\{\phi\}, \emptyset, \emptyset, \emptyset)$
end

And the value of v determines whether ϕ is **K**-satisfiable

K-WORLD Algorithm

Next we examine the space complexity of the algorithm.
The recursive nature of the algorithm is implemented on a Turing machine by simulating a stack. Since at each level, the four sets will only contain subformulas of ϕ, hence they can be represented "pointers" to ϕ.
There are four different marks for each of the four sets. To implement the pointer, copy the original formula and put the marks on the major connectives of each subformula and each propositional letters to indicate whether the subformula belongs to the set.
The way we implement the pointer implies that the storage requirments on each level of recursion is $O(n)$.
When does the requirement for storage space increase?

K-WORLD Algorithm

Then we need to show that the number of recursion is also $O(n)$

Proof.

For a finite set of formulas Σ, we define $|\Sigma| = \sum_{\phi \in \Sigma} |\phi|$. Then by induction on $n = |\Sigma_1| + |\Sigma_2| + |\Sigma_3| + |\Sigma_4|$,we can show that the number of recursion of *K-WORLD*$(\Sigma_1, \Sigma_2, \Sigma_3, \Sigma_4)$ is at most $2n + 1$:

$n = 0$: The situation here is trival

$n > 0$: Suppose first recursion call is *K-WORLD*$(\Sigma_1, \Sigma_2, \Sigma_3, \Sigma_4)$ to *K-WORLD*$(\Sigma_1', \Sigma_2', \Sigma_3', \Sigma_4')$. If $\Sigma_1 \cup \Sigma \neq \emptyset$, we can check case by case to see that $|\Sigma_1'| + |\Sigma_2'| + |\Sigma_3'| + |\Sigma_4'| = n' \leq n - 1$, by induction hypothesis we have that *K-WORLD*$(\Sigma_1', \Sigma_2', \Sigma_3', \Sigma_4')$ has at most $2n' + 1$ level of recursion. Otherwise, we must be at the line where $\Sigma_1 \cap \Sigma_2 = \emptyset$, then it will reduct to the situation above. Hence two level reduce the total length for at least 1, and the

2.Ladner's Theorem

Preparations for Ladner's theorem

Definition

The set of *quantified boolean formulas* is the smallest set X containing all formulas of propositional calculus such that

$$\beta \in X \text{ and } p \text{ is a propositional letter } \Longrightarrow \forall p\beta \in X, \exists p\beta \in X$$

A quantified boolean formula is called QBF if it is in the following form

$$Q_1 p_1 \dots Q_m p_m \theta(p_1, \dots, p_m)$$

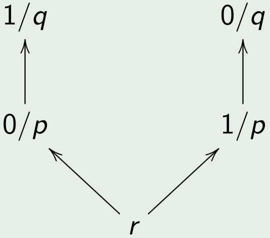
where $Q_i = \forall$ or $\exists$ and θ is a formula of propositional logic.

The problem of deciding whether a QBF is valid is called *QBF-vadility problem*

Preparations for Ladner's theorem

Example

An annotated tree to evaluate a QBF $\forall p \exists q (p \leftrightarrow \neg q)$



Preparations for Ladner's theorem

Theorem

The QBF-vadility problem is PSPACE-hard

Proof.

See Papadimitriou [356, Theorem 19.1]. □

Preparations for Ladner's theorem

Definition

Define the following formulas for $\beta = Q_1 p_1 \dots Q_m p_m \theta(p_1, \dots, p_m)$ with new variables $q_0, \dots, q_m$

(i) q_0

(ii) $\Box^{(m)} (q_i \to \Lambda_{i \neq j} \neg q_j) \quad (0 \leq i \leq m)$

(iiia) $\Box^{(m)} (q_i \to \Diamond q_{i+1}) \quad (0 \leq i < m)$

(iiib) $\bigwedge_{\{i | \mathrm{Q}_i = \forall\}} \Box^i B_i$

(iv)

$$\begin{aligned}
\Box S (p_1, \neg p_1) \wedge \Box^2 S (p_1, \neg p_1) \wedge \Box^3 S (p_1, \neg p_1) \wedge \cdots \wedge \Box^{m-1} S (p_1, \neg p_1) \\
\wedge \Box^2 S (p_2, \neg p_2) \wedge \Box^3 S (p_2, \neg p_2) \wedge \cdots \wedge \Box^{m-1} S (p_2, \neg p_2) \\
\wedge \Box^3 S (p_3, \neg p_3) \wedge \cdots \wedge \Box^{m-1} S (p_3, \neg p_3) \\
\vdots \\
\wedge \Box^{m-1} S (p_{m-1}, \neg p_{m-1})
\end{aligned}$$

(v) $\Box^m (q_m \to \theta)$

Preparations for Ladner's theorem

Definition

Given $\beta = Q_1 p_1 \ldots Q_m p_m \theta(p_1, \ldots, p_m)$, $f_L(\beta)$ is the conjuction of the formulas defined on last page.

Note that $f_L(\beta)$ is polysize in $|\beta|$.

Ladner's theorem

Theorem

If Λ is a normal modal logic such that **K** $\subseteq \Lambda \subseteq$ **S4**, *then Λ has a PSPACE-hard satisfiability problem. Moreover, Λ has a PSPACE-hard validity problem.*

The general idea to prove the theorem is to show that f_L is a reduction from QBF-vadility problem to Λ-satisfiability problem. To show this, there are two crucial statement to bridge the gap:

(i) if β is a QBF-validity, then $f_L(\beta)$ is satisfiable on a frame for **S4**
(ii) if $f_L(\beta)$ is satisfied in a **K**-model then β is a QBF-validity.

So it remains to prove the two statement.

Proof.

For statement (i):
Assume that β is a QBF-vadility. Then we can generate a quantified tree witnessing the vadility of β. Then we can construct an **S4**-model $\mathcal{M}$ from the tree: let $W_{\mathcal{M}}$ be the set of nodes of the tree, then take $R_{\mathcal{M}}$ to be the transitive and reflexive closure of the 'daughter of' relation, this give us the **S4**-frame; then define V that make q_i true precisely at nodes of level i, and p_i is made true on a node of level $j \geq i$ iff the substitution connected to that node or its predecessors at level i returns the value 1 for p_i. Then we need to check that $\mathcal{M}, r \vDash f_L(\beta)$

Ladner's theorem

Proof.

We need to check whether $\mathcal{M}, w$ satisfy each part of the formula. Items (i) (ii) and (iiia) are easy to check. For item (iiib), given an i such that $Q_i = \forall$, by definition of the model, a node w at level i must have two successors w_1, w_2 such that p_{i+1} is made true on only one of them(say w_1). Hence we have that $\mathcal{M}, w_1 \vDash q_{i+1} \wedge p_{i+1}$ and $\mathcal{M}, w_2 \vDash q_{i+1} \wedge \neg p_{i+1}$, hence $\mathcal{M}, w \vDash B_i$, and hence $\mathcal{M}, r \vDash \Box^i B_i$.

For (iv), we just show the first line and the rest is analogue. By the definition of V, p_1 is made true on a node w in level $i, i > 1$ iff p_i is made true on the predecessor in level 1, and so does any successor of w in level $i+1$, hence we have that $\mathcal{M}, w \vDash S(p_1, \neg p_1)$. Hence we have that $\mathcal{M}, r \vDash \Box^i S(p_1, \neg p_1)$. And the last item is satisfied by the definition of the quantifier tree.

Proof.

For statement (ii):
Suppose that the quantifier degree of β is m and $f_L(\beta)$ is **K**-satisfiable. Note that $\deg(f_L(\beta))$ is m, hence from the proof of Lemma 6.46(the blue book), we know that $f_L(\beta)$ is satisfiable holds at the root r of a tree-based model $\mathcal{M} = (T, R, V)$ of depth at most m. Using (iiia) and (iiib) in the definition of $f_L(\beta)$, it is easily verified that we may cut off branches from this tree such that in the resulting tree, a node at level $i < m$ has either one or two successor. The number is one iff $Q_i = \exists$, if $Q_i = \forall$, the number is two and one of them satisfies p_i while the other satisfies $\neg p_i$. Ans that is a witnessing tree for vadility of β □

References

[1] Ladner, R. E .*The Computational Complexity of Provability in Systems of Modal Propositional Logic*[J]. SIAM Journal on Computing, 1977, 6(3):467-480.
[2] Blackburn, P., de Rijke,M., Venema,Y. *Modal Logic*[M],2001