An Introduction to Modal Logic X

PSPACE hardness

Marco Cerami

Palacký University in Olomouc
Department of Computer Science
Olomouc, Czech Republic

Olomouc, November 28^{th} 2013

european social fund in the czech republic
EUROPEAN UNION
MINISTRY OF EDUCATION, YOUTH AND SPORTS
OP Education for Competitiveness
INVESTMENTS IN EDUCATION DEVELOPMENT

Introduction

- We have proved that the minimal normal modal logic K is **decidable**, but the size of its **models can be larger than polynomial** on the size of the formulas;
- this means that $K \notin NP$.
- so, normal modal logics are inherently **intractable**.
- But, **how intractable they are**?
- Normal modal logic belongs to a **wide range of complexity classes**, from NP up to those that are undecidable;
- Here **we will put our attention on** the minimal modal logic K that is PSPACE-complete.
- Before proving this fact, we are providing a brief reminder about the class PSPACE and its most representative problem: **Quantified Boolean Formulas**.

The class PSPACE

The class PSPACE

- The complexity class PSPACE is the class of all those problems that are solvable by a **deterministic** Turing machine using an **amount of space** that is polynomial on the size of the input instance.

- Some of the proved properties of the class PSPACE are the following:
  - $\mathrm{PSPACE} = \mathrm{NPSPACE}$;
  - $\mathrm{PSPACE} =$ co-PSPACE;
  - $\mathrm{NP} \subseteq \mathrm{PSPACE}$;
  - $\mathrm{PSPACE} \subseteq \mathrm{EXPTIME}$;
  - it is still an **open problem** whether $\mathrm{PSPACE} \nsubseteq \mathrm{NP}$;

Quantified Boolean Formulas

- Let φ be a **propositional formula** with variables in $\{p_1, \dots p_n\}$;
- let $Q_1, \dots, Q_n \in \{\forall, \exists\}$ be quantifiers **ranging over** $\{0, 1\}$.
- A**quantified boolean formula** is an expression of the form:

$$Q_1 p_1 \dots Q_n p_n \varphi(p_1, \dots, p_n)$$

- **QBF** is the set of all quantified boolean formulas;
- the problem of deciding whether a quantified boolean formula is true is called the **QBF truth problem** (we will call it QBF for short).
- QBF is known to be a PSPACE-**complete problem**.

Some remarks on QBF

- Informally, a qbf of the form:

$$\exists p \forall q (p \to q)$$

  means "there exists a truth assignment in $\{0,1\}$ to p such that, for every truth assignment in $\{0,1\}$ to q, the formula $p \to q$ evaluates to 1".

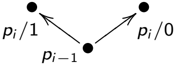
- This example is indeed a **true instance** of QBF, since, if we assign truth value 0 to variable p, then the implication $p \to q$ is true for every assignment to q.
- It is easy to see that a qbf where **just existential quantifiers** appear, is an instance of the SAT problem;
- In the same way, a qbf where **just universal quantifiers** appear, is an instance of the propositional validity problem;

Evaluating quantified boolean formulas

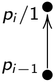
- In this sense, a qbf **is not just** either satisfiable or valid and we prefer to say that it is “true” (or not);
- The process of evaluating a qbf, can be represented by **quantifier trees**.
- Quantifier trees, are similar but different from he evaluation trees typical in SAT, since they **are not binary trees**.

Quantifier trees

- The **rules for building quantifier trees** for a formula $Q_1 p_1 \dots Q_n p_n \varphi(p_1, \dots, p_n)$ are as follows:
  1. add a **root node** that refers to the whole formula $Q_1 p_1 \dots Q_n p_n \varphi(p_1, \dots, p_n)$;
  2. at each step consider the **next propositional letter** p_i in the list $Q_1 p_1 \dots Q_n p_n$ and proceed as follows:
     - ★ if $Q_i = \forall$, then add **two** edges connecting to two nodes and label one node with 0 and the other with 1,



     - ★ if $Q_i = \exists$, then add **one** edge connecting to one node and label the node with either 0 or 1;



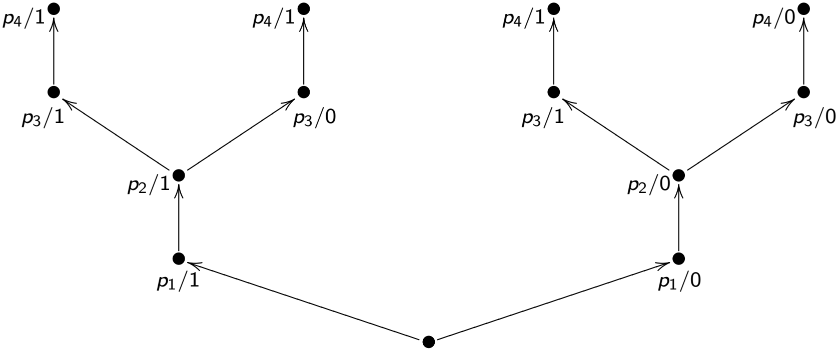
  3. proceed until p_n is processed.

Quantifier trees: an example

Consider the quantified boolean formula:

$$\forall p_1 \exists p_2 \forall p_3 \exists p_4 ((p_1 \to p_2) \vee (p_3 \wedge p_4))$$

Hence, its quantifier tree should have this form:



The logic K is PSPACE-hard

Proving PSPACE-hardness

- As usual, the hardness proof for the satisfiability problem of K is obtained **by polynomial reduction** of a PSPACE-complete problem.
- In our case the PSPACE-complete problem to be reduced is **QBF**.
- The **original proof** is by Ladner, 1977,
- In the original proof it is proved PSPACE-hardness **for all normal modal logics between** K **and** $S4$.
- Despite the fact that the result is proved for the satisfiability problem, it is easily obtained **also for the validity problem**.

Reduction of QBF to K

Consider any quantified boolean formula

$$\beta = Q_1 p_1 \dots Q_n p_n \varphi(p_1, \dots, p_n)$$

and choose **new** propositional variables $q_0, \dots, q_n$, then $f(\beta)$ is the conjunction of the following formulas:

$$
\begin{array}{lll}
\text{(i)} & q_0 & \\
\text{(ii)} & \Box^{(n)}(q_i \to (\wedge_{i \neq j} \neg q_j)) & (0 \leq i \leq n) \\
\text{(iiia)} & \Box^{(n)}(q_i \to \Diamond q_{i+1})) & (0 \leq i < n) \\
\text{(iiib)} & \bigwedge_{\{i \mid Q_i = \forall\}} \Box^i B_i & \\
\text{(iv)} & \begin{array}{lllll}
\Box S_1 & \wedge \Box^2 S_1 & \wedge \Box^3 S_1 & \wedge \dots \wedge & \Box^{n-1} S_1 \\
 & \wedge \Box^2 S_2 & \wedge \Box^3 S_2 & \wedge \dots \wedge & \Box^{n-1} S_2 \\
 & & \wedge \Box^3 S_3 & \wedge \dots \wedge & \Box^{n-1} S_3 \\
 & & & & \vdots \\
 & & & & \wedge \Box^{n-1} S_{n-1}
\end{array} & \\
\text{(v)} & q_n \to \varphi &
\end{array}
$$

where, again, for every i with $0 \leq i \leq n-1$ we keep the following formulas

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

and

$$S_i \quad := \quad (p_i \rightarrow \Box p_i) \wedge (\neg p_i \rightarrow \neg \Box p_i)$$

and the following abbreviations:

$$\Box^i \psi := \overbrace{\Box \ldots \Box}^{i \text{ times}} \psi \qquad \text{and} \qquad \Box^{(m)} \psi := \psi \wedge \Box \psi \wedge \Box^2 \psi \wedge \ldots \wedge \Box^m \psi$$

Every model of $f(\beta)$ contains a quantifier tree

Again, each model of $f(\beta)$ contains a quantifier tree of depth n:

1. item (i) forces for a root node,
2. item (ii) forces that in in every node only one among $q_0, \ldots, q_n$ is true,
3. item (iiia) forces that, at level i, every node has at least **one successor**, where the value of p_i is left undetermined,
4. item (iiib) forces that, at level i, every node that is followed by a **universally quantified variable** has at least two successors, one with p_i and the other with $\neg p_i$,
5. item (iv) forces the propagation of either p_i or $\neg p_i$ to every path that starts from a point where either p_i or $\neg p_i$ are true,
6. finally, item (v) forces that φ has to be true **in every leaf node** and **with each propagated evaluation of** $p_1, \ldots, p_n$.

If β is true, then $f(\beta)$ is satisfiable

1. Suppose that β is true,
2. then there exists a quantifier tree that is a model of β,
3. from this quantifier tree build a frame $\mathfrak{F} = \langle W, R \rangle$ where
   - W is the set of nodes of the tree,
   - R is defined following the edges of the tree.
4. On $\mathfrak{F}$ define the valuation V in the following way:
   - give value 1 to variable q_i at every node in level i and 0 otherwise,
   - follow the valuation of the tree to evaluate variables $p_1, \ldots, p_n$.
5. It is easy to see that $\mathfrak{M} = \langle W, R, V \rangle$ satisfies $f(\beta)$ at the root node.

If $f(\beta)$ is satisfiable, then β is true

1. Suppose that $f(\beta)$ is satisfiable,
2. then there exists a model $\mathfrak{M} = \langle W, R, V \rangle$ that satisfies $f(\beta)$ at some node w,
3. as we have seen, $\mathfrak{M}$ contains a quantifier tree whose root is w,
4. select just this tree and substitute the accessibility relation by edges,
5. follow the valuation V of the variables $p_1, \ldots, p_n$ to evaluate the same variables in β when they are under the direct scope of an existential quantifier,
6. it is easy to see that the valuation so obtained makes true β

Conclusion of the proof

- As we have seen, the quantified boolean formula β is true if and only is the modal formula $f(\beta)$ is satisfiable,
- moreover, as we have seen for the formula $\varphi^{\mathcal{B}}(n)$, we have that the size of $f(\beta)$ is polynomial in the number of propositional variable appearing in β
- hence, $f(\beta)$ is at most polynomial in the size of β.
- As a consequence, the satisfiability problem for K is PSPACE-hard.

Further consequences

We have a couple of further remarks about the proof and the result:

- In the original proof by Ladner, it is proved that **every logic between K and $S4$** is PSPACE-hard.
- To obtain this result, it is **enough to change the first part** of the proof and proving that if β is true, then $f(\beta)$ is satisfiable in a $S4$-model.
- Obtaining an $S4$-model is easy, it is enough to **add all the relations to the frame** that we have defined from the quantifier tree in such a way that the **result is a transitive-reflexive frame**.
- Finally, since PSPACE=co-PSPACE, we have that the **validity problem** of any normal modal logic between K and $S4$ is PSPACE-hard.