The Complexity of Model Checking: Preprocessing and Succinctness

Andrea Ferrara, Paolo Liberatore, Marco Schaerf

Dipartimento di Informatica e Sistemistica

Università di Roma "La Sapienza"

The Model Checking Problem

input: a finite structure P (the model) and a formula φ

question: P is a model of φ ?

In *verification* [CGP00] P is a concurrent transition system and φ is a temporal logic formula

The Model Checking Problem (2)

- P represents a discrete system (e.g. a set of semaphores)
  - it has n components $P_1, \ldots, P_n$
  - each component represents a component of the system (e.g. a circuit, an agent, ..)
  - all components are executed concurrently

- φ represents a property that the system must satisfy; in this work we assume that φ is a formula in Linear-Time Temporal Logic (LTL), one of the most used

Transition Systems

- a labelled transition system (LTS) is a valued, oriented graph whose nodes are called states, and whose arcs are called transitions
- each state is labelled by the set of atomic propositional symbols true in the state

Concurrent Transition System

A concurrent LTS has n LTS (components) running concurrently

- its set of states W is the union of the set of states of all components
- the transition relation of the component i-th is a relation $\delta_i \subset (W_i \times \beta(W) \times W_i)$, where W_i is the set of states of the component i-th, and $\beta(W)$ the set of Boolean formulae on the variables in W
- example: let a transition be, in the component P_i, (s_1^i, θ, s_2^i) where $theta = s_3^2 \wedge s_4^3$. It means that there is a transition from the state s_1^i to the state s_2^i if the Boolean formula θ is satisfied by the concurrent LTS, that is if the component P_2 is in the state s_3 and the component P_3 is in the state s_4)

Linear-Time Temporal Logic

- propositional logic plus temporal operators

- temporal operators:
  - $X\alpha$ is true iff α is true in the next state
  - $F\alpha$ is true iff α is true in one of the following state
  - $G\alpha$ is true iff α is true in all the following state
  - $\alpha\ U\ \beta$ is true iff α is true until β becomes true

The Complexity of Model Checking

- the Model Checking problem is PSPACE-complete [KVW00]

- many tools allow a preprocessing step either for the model either for the formula

- allowing this preprocessing step, is there a polynomial time algorithm?

The Complexity of Model Checking allowing a preprocessing step

- we cannot store an exponential amount of information after the preprocessing step (impossible in practice)

- the result of the preprocessing must be stored in polynomial space respect the input

- we use the compilability theory

The Compilability Theory

- defined in [CDLS02, Lib01, CDLS99, CDLS00]
- it defines the complexity of problems with two inputs
- one input (the fixed part) is compiled in a polynomial data structure
- the other input is given on-line

The Compilability Theory (2)

It defines:

- compilability complexity classes for this kind of problems (e.g. $\|\!\!\leadsto$NP, $\|\!\!\leadsto$PSPACE)

- the notion of poly-size reduction, that is, a reduction whose result is poly-size respect to the input

- the notion of completeness of a compilability complexity class respect to the poly-sized reduction

Preprocessing the Model Checking - Results

- The Model Checking problem, preprocessing in a polynomial size data structure either the model or the formula is $\Vdash\!\!\leadsto$PSPACE

- a consequence: there is no polynomial algorithm even with a preprocessing step

Symbolic Model Checking

- a concurrent transition system with n component $P_1, \cdots, P_n$ has an equivalent transition system (non concurrent) of size $|P_1| \cdot |P_2| \cdot \cdots \cdot |P_n|$ [HKV02]

- a concurrent transition system is a succinct representation of the model

- all model checking algorithms that represents explicitly the states suffers of this state-explosion problem

- there are algorithms that work on symbolic representation of the states (using Boolean funcions)

Symbolic Model Checking (2)

- the most useful representation for manipulating Boolean functions are BDD (Binary Decision Diagrams) [DS01]

- OBDDs (Ordered BDD) suffer of the state-explosion problem: in [Bry91] it is shown that the multiplier circuit cannot be represented in polynomial space by an OBDD, for any variable order

- other Decision Diagrams can represent the multiplier in polynomial space [DS01]

Succinctness of data structures for Symbolic Model Checking

- are there data structures that allow a Symbolic Model Checking algorithm to run in polynomial space respect the size of the concurrent LTS and the formula ?

- we can use the compilability theory to estabilish it

Compilability and Succintness

- if a problem p is in $\|\!\!\leadsto$C, if a problem p' is in $\|\!\!\leadsto$C$'$-complete, if $C \subset C'$, then there is not a poly-size reduction from p to p'

- we exploit this in the proof of the following theorem

Compilability and Succintness - theorem

Theorem : if $\Sigma_i \subset \Sigma_{i+1}$, all Symbolic Model Checking algorithm have to use more than polynomial space respect to the size of the concurrent transition system and the formula, for any data structure whose evaluation problem is in the polynomial hierarchy.

evaluation problem: estabilishing the value of the data structure for a particular input (in the case of BDD the input is an assingment, and we evaluate the assigment)

Compilability and Succinctness - discussion

Respect [Bry91]

- we show a more general result: the result in [Bry91] only deals with OBDDs

- we have a conditional result (the polynomial hierarchy does not collapse); in [Bry91] there are unconditional results

Conclusion

We showed two negative results about the complexity of Model Checking problem:

- even preprocessing either for the model or for the formula and storing the result in polynomial space, the complexity of the Model Checking problem is PSPACE-complete

- if the polynomial hierarchy does not collapse, Symbolic Model Checking algorithms have to use more than polynomial space, for any data structures that they use

Future Work

- the complexity of Model Checking under restrictions, both structural and syntactical

References

R.E. Bryant. On the complexity of VLSI implementations and graph representations of boolean functions with application to integer multiplication. *IEEE Trans. on Computers*, 40:205–213, 1991.

M. Cadoli, F.M. Donini, P. Liberatore, and M. Schaerf. Space efficency of propositional knowledge representation formalisms. *JAIR*, 13:25–64, 1999.

M. Cadoli, F.M. Donini, P. Liberatore, and M. Schaerf. The size of a revised knowledge base. *Art. Int.*, 115:1–31, 2000.

M. Cadoli, F.M. Donini, P. Liberatore, and M. Schaerf. Preprocessing of intractable problems. *Inf. and Comp.*, 176:89–120, 2002.

E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking*. MIT Press, 2000.

R. Drechsler and D. Sieling. Binary decision diagrams in theory and practice. *STTT*, 3(2):112–136, May 2001.

D. Harel, O. Kupferman, and M.Y. Vardi. On the complexity of verifying concurrent transition systems. inf. comput. 173(2): 143-161 (2002). *Information and Computation*, 173(2):143–161, 2002.

O. Kupferman, M.Y. Vardi, and P. Wolper. An automata theoretic approach to branching-time model checking. *Journal of ACM*, 47(2):312–360, 2000.

P. Liberatore. Monotonic reductions, representative equivalence, and compilation of intractable problems. *Journal of ACM*, 48(6):1091–1125, 2001.