

# Implementation of propositional temporal logics using BDDs

**Citation for published version (APA):**

Janssen, G. L. J. M. (1998). Implementation of propositional temporal logics using BDDs. In H. Swart, de (Ed.), *TABLEAUX, automated reasoning with analytic tableaux and related methods : international conference, 1998, Oisterwijk, The Netherlands, May 5-8, 1998 : proceedings* (pp. 40-41). (Lecture Notes in Computer Science; Vol. 1397). Springer. [https://doi.org/10.1007/3-540-69778-0\\_11](https://doi.org/10.1007/3-540-69778-0_11)

**DOI:**

[10.1007/3-540-69778-0\\_11](https://doi.org/10.1007/3-540-69778-0_11)

**Document status and date:**

Published: 01/01/1998

**Document Version:**

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

**Please check the document version of this publication:**

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

**Take down policy**

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

# Implementation of Propositional Temporal Logics Using BDDs

G.L.J.M. Janssen

Eindhoven University of Technology  
Department of Electrical Engineering, Room EH 9.26  
P.O. Box 513, 5600 MB Eindhoven  
E-mail: geert@ics.ele.tue.nl

## 1 Topic/Relevance

This tutorial intends to convey the step-by-step process by which computer programs for model checking and satisfiability testing for temporal logics may be derived from the theory. The idea is to demonstrate that it is very well possible to implement such a program in an efficient way without sacrificing a correct-by-construction approach. The tutorial will be fully self-contained, only a general knowledge of programming and propositional logic is assumed.

The proposed tutorial focuses on what might be coined “Implementing the theories”. Unfortunately, too often an interesting approach or novel algorithm once published is not picked up by any user community. This is partly because no effort is spent in creating a state-of-the-art implementation in a readily accessible form. The intended audience are researchers in the field of theorem proving and any users of applications thereof. They will benefit from learning how theoretical results can be implemented in a well-written prototype program. Moreover, being a prototype should not be an excuse for not using advanced datastructures and algorithms. Therefore this tutorial takes an engineering approach to dealing with complicated issues such as reasoning in temporal logic, and shows that with a structured approach a powerful tool can be built that is able to handle real-life applications, for instance verification of sequential circuits.

## 2 Contents Outline

Below an outline is given of the tutorial contents. Much of the material on CTL model checking is based on the pioneering work in this area by Clarke et al.

1. **Introduction.** Presents the tutorial contents and sets its goals.
2. **Dags.** Directed acyclic graphs are the datastructure underlying BDDs. Dags will also be used to represent formulas. An efficient implementation based on a hash table and utilizing garbage-collection will be discussed.
3. **BDDs.** Explains what Binary Decision Diagrams are, how they can be efficiently implemented, and what their applications are.
4. **Kripke Structure Model.** Introduces the notion of a Kripke structure that is used as a model for the system we like to reason about.

5. **Computation Tree Logic.** Defines the syntax and semantics of a class of branching time logics.
6. **CTL Model Checking.** Shows how a model checker for CTL can be constructed in elegant way using a simple algorithm to calculate a fixed-point of a functional. Also, it is shown how BDDs can be exploited to represent the next-state relation and state-sets of the Kripke structure.
7. **Linear-time Temporal Logic.** Defines the syntax and semantics of the popular Manna/Pnueli propositional linear-time temporal logic.
8. **PTL Satisfiability Checking.** Shows how a satisfiability checker can be constructed for PTL. Again, BDDs will be used to represent the structure of the generated tableau.
9. **Diversion into  $\mu$ -calculus.** Briefly explains Kozen's  $\mu$ -calculus and shows how both CTL and PTL problems can be encoded in it. An implementation of  $\mu$ -calculus is sketched.
10. **Summary and conclusions.** Summarizes the presented material and demonstrates the derived programs by some example runs and lists results of experiments.

The bibliography lists a number of articles and books on which the tutorial material will be based.

## References

1. Z. Manna, A. Pnueli, "Verification of concurrent programs: the temporal framework," *The Correctness Problem in Computer Science*, eds. Robert S. Boyer, J. Strother Moore, International Lecture Series in Computer Science, Academic Press, New York, 1981.
2. J.R. Burch, E.M. Clarke, K.L. McMillan, "Symbolic Model Checking:  $10^{20}$  States and Beyond," *International Workshop on Formal Methods in VLSI Design*, 1991.
3. E. Clarke, O. Grumberg, K. Hamaguchi, "Another Look at LTL Model Checking," *Proceeding 6-th Int. Conf. on Computer Aided Verification*, Springer-Verlag, LNCS 818, ed. David L. Dill, pp. 415–427.
4. Geert L.J.M. Janssen, "Hardware Verification using Temporal Logic: A Practical View," *Formal VLSI Correctness Verification, VLSI Design Methods-II*, Elsevier Science Publishers B.V. (North-Holland), ed. L.J.M. Claesen, IFIP, 1990, pp. 159–168.
5. Randal E. Bryant, "Graph-Based Algorithms for Boolean Function Manipulation," *IEEE Transactions on Computers*, Vol. C-35, Nr. 8, August 1986.
6. Karl S. Brace, Richard L. Rudell, Randal E. Bryant, "Efficient Implementation of a BDD Package," *Proceedings Design Automation Conference*, June 1990.
7. Kenneth L. McMillan, *Symbolic Model Checking*, Kluwer Academic Publishers, 1993.
8. E.A. Emerson, "Chapter 16: Temporal and Modal Logic," *Handbook of Theoretical Computer Science*, B: Formal Models and Semantics, ed. Jan van Leeuwen, Elsevier Science Publishers B.V., 1990, pp. 996–1072.
9. Rance Cleaveland, "Tableau-Based Model Checking in the Propositional  $\mu$ -Calculus," *Acta Informatica*, No. 27, 1990, pp. 725–747.