Chapter 13
Symbolic Model Checking for Interlocking Systems

Kirsten Winter
The University of Queensland, Australia

ABSTRACT

Model checking is a fully automated technique for the analysis of a model of a system. Due to its degree of automation it is in principle suitable for application in industry but at the same time its scalability is limited. Symbolic model checking is one approach that improves scalability through the use of Binary Decision Diagrams (BDDs) as an internal data structure. This approach allows the user to increase the efficiency by customising the ordering of state variables occurring in the model to be checked. In the domain of railway interlockings represented as control tables, it is found that this task can be supported using an algorithm that has access to the track layout information. In our work we propose optimisation strategies that render symbolic model checking feasible for large scale interlocking systems.

Our results yield a verification tool suitable for use in industry.

INTRODUCTION

Railway signalling interlockings are safety critical systems. They are designed to permit the safe movement of trains along a railway system. Therefore special attention has to be given to the correctness of the design and the implementation of an interlocking system. In order to guarantee a safe functioning of the system railway engineers are currently manually validating the Interlocking against the Signalling Principles.

The development of such systems is very labour intensive and prone to error. It requires specialised skills. Moreover, possible errors in the design are detected very late in the design process. To mitigate these problems Queensland Rail (QR), the major railway operator and owner in Queensland, Australia, intended to support its design process by a specialised tool set called the Signalling Design Toolset (SDT) (which was first introduced in Robinson, Barney, Kearney, Nikandros & Tombs (2001)). Parts of this toolset were intended for supporting the verification task.
Symbolic Model Checking for Interlocking Systems

Railway interlockings, next to hardware designs, have been shown earlier to be a suitable application for automated verification techniques, in particular model checking and automated theorem proving (Groote, Koorn & van Vlijmen, 1995; Eisner, 1999; Boralv & Stålmarck, 1999; Huber & King, 2002; Simpson, Woodcock & Davies, 1997). In our work we propose to integrate model checking into the design process to ensure the correctness of a design before it is implemented.

Model checking is a fully automated technique for the analysis of a model of a system. Due to its degree of automation it is suitable for application in industry but at the same time its scalability is limited. For complex system models checking the complete state space often leads to the state explosion problem, a situation where run-time and memory usage of the checking process exceed the tolerable limits (as too many states are to be explored). This problem has been targeted by researchers in the last twenty years. The aim is to improve the efficiency of the algorithms and push the boundary of feasible models that can be checked automatically.

Symbolic model checking (McMillan, 1993; Burch, Clarke, McMillan, Dill & Hwang, 1992) is one of the early proposals for improving the efficiency of model checking. In this approach the model, its states and transitions between states (i.e. its behaviour) are represented by a data structure called Binary Decision Diagrams, BDDs (Bryant, 1986). This data structure has the advantage that in some cases its representation can be reduced to a structure of very small size. Consequently, less memory space is necessary and the operations to compute and check the state space become very fast. What is needed, however, is a good ordering of the variables that describe the model to allow for a sufficient - if not optimal - reduction of the BDDs.

Queensland Rail (QR) contracted the Software Verification Research Centre in 2002 to conduct a feasibility study on the automated analysis of control tables. Control tables instantiate the generic signalling and operational principles for interlocking systems in a particular country or region (Queensland Rail Signal and Operational Systems, 1998). Each instantiation is given with respect to a particular geographical layout of the railway equipment. We performed this research in close cooperation with the railway engineers. The results of the feasibility study (Winter, 2002a) were in favour of the approach of symbolic model checking and using the tool NuSMV (Cimatti, Clarke, Giunchiglia, Giunchiglia, Pistore, Roveri, Sebastiani & Tacchella, 2002). They support the claim that symbolic model checking scales sufficiently well if enhanced by a good user-defined variable ordering. For control tables a very good ordering can be generated from the layout information that is available. An algorithm that can process this layout information can generate the ordering and therefore, once set up, the process is fully automatic. The improvement of efficiency is for some models more than ten-fold.

Others have applied model checking to analyse railway interlocking systems: Gnesi, Lenzini, Latella, Abbaneo, Amendola & Marmo (2000), Bernardeschi, Fantechi, Gnesi & Mongardi (1996), and Cleaveland, Luettgen & Natarajan (1996), for instance, have analysed fault tolerance of interlocking systems. The main focus in their work was communication aspects between components rather than the control logic as in our work.

The first approaches on applying model checking to verify railway interlocking systems represented as control tables or equations were reported by Groote, Koorn & van Vlijmen (1995) using µCRL and its tools, and Eisner (1999; 2002) using SMV. Also Simpson, Woodcock & Davies (1997), and Huber & King (2002) are focusing on the control logic similar to our approach. In particular, the approach by Eisner (1999; 2002) and Huber & King (2002) is very close to ours as they also use a symbolic model checker. The interlocking systems, however, are modelled on a lower level of abstraction. This leads to significantly different models and in particular a more
16 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product's webpage:
www.igi-global.com/chapter/symbolic-model-checking-interlocking-systems/66677?camid=4v1

www.igi-global.com/e-resources/library-recommendation/?id=1

Related Content

**Execution Management for Mobile Service-Oriented Environments**
www.igi-global.com/article/execution-management-mobile-service-oriented/47037?camid=4v1a

**Design of the While Listening Activities in Interactive Multimedia Listening Software**
www.igi-global.com/chapter/design-of-the-while-listening-activities-in-interactive-multimedia-listening-software/188235?camid=4v1a

**A Structured Method for Security Requirements Elicitation concerning the Cloud Computing Domain**
www.igi-global.com/article/a-structured-method-for-security-requirements-elicitation-concerning-the-cloud-computing-domain/113725?camid=4v1a

**Agile Software Methods: State-of-the-Art**
www.igi-global.com/chapter/agile-software-methods/5066?camid=4v1a