Validation of an Interlocking System by Model-Checking

Andrea Bonacchi
DINFO – University of Florence
Via Santa Marta, 3 Firenze, Italy
a.bonacchi@unifi.it

Abstract - Railway interlocking systems still represent a challenge for formal verification by model checking: the high number of complex interlocking rules that guarantee the safe movements of independent trains in a large station makes the verification of such systems typically incur state space explosion problems.
We describe a study aimed to define a verification process based on commercial modelling and verification tools, for industrially produced interlocking systems, that exploits an appropriate mix of environment abstraction, slicing and CEGAR-like techniques, driven by the low-level knowledge of the interlocking product under verification, in order to support the final validation phase of the implemented products.

In the railway signalling domain, an interlocking is the safety critical system that controls the movement of the trains in a station and between adjacent stations. The interlocking monitors the status of the objects in the railway yard and allows or denies the routing of the trains in accordance with the railway safety and operational regulations that are generic for the region or country where the interlocking is located. The instantiation of these rules on a station topology is stored in the part of the system named control table. Control tables of modern computerized interlockings are implemented by means of iteratively executed software controls over the status of the yard objects.
One of the most common way to describe the interlocking rules given by control tables is through Boolean equations or, equivalently, ladder diagrams which are interpreted either by a PLC or by a proper evaluation engine over a standard processor.
Verification of correctness of control tables has been a prolific domain for formal methods practitioners. Model checking in particular has raised the interest of many railway signalling industries, being the most lightweight from the process point of view, and being rather promising in terms of efficiency: safety properties of an interlocking system are quite directly expressed in temporal logic, and their specifications by means of control tables can be directly formalized. However, due to the high number of Boolean variables involved, automatic verification of sufficiently large stations typically incurs in combinatorial state space explosion problem.
The first applications of model checking have attacked portions of an interlocking system; but even recent works show that routine verification of interlocking designs for large stations is still out of reach for symbolic model checker NuSMV and explicit model checker SPIN. One point in common to more recent works, addressing different verification aims, is the use of SAT-based model checking, which appears to be more promising at this respect.
A joint project with our industrial partner has defined a validation activity, aiming at reducing the costs of verifying the safety requirements of the produced interlocking systems, in the

system validation phase. This validation activity considers the control tables as implemented in the produced interlocking systems, and extracts from legacy control tables a Simulink model made of Boolean functions with logical gates.
The obtained Simulink model is used in the daily verification activity of our industrial partner to simulate on the model the same test cases foreseen for the produced system, in order to profit of the radically (up to twenty times) shorter time w.r.t. testing.
We then set up a verification framework based on model checking on the extracted model, employing Matlab Design Verifier, both since it works on Simulink models and to exploit at best its SAT-solving capabilities on the native Boolean coding of the control tables. The verification framework aims to make more precise the kind of verification process that the framework conveniently supports, by considering an appropriate use of environment abstraction, slicing and CEGAR-like techniques, driven by the detailed knowledge of the interlocking product under verification. In particular, an iterative verification process has been defined, which starts by considering a *slice* of the model that includes only those Boolean equations that have a direct or indirect impact on the outputs observed by the property to be checked. If the slice is not sufficient to prove the property, other equations are added to the slice, or the verification depth is increased. Indeed the adopted model checker, Design Verifier, is a SAT-based Bounded Model Checker, and hence it is best aimed at verifying that the Boolean output of the observer holds for all the states up to a given depth. However, Design Verifier has not allowed to play with different verification strategies being the tool a commercial, quite closed one.
The conducted verification experiments have shown the feasibility of the proposed iterative approach on slices of models derived by an industrial design of a quite large sized interlocking. Indeed, the whole model consisting of one thousand equations is actually not easy to deal with, although Design Verifier was able to terminate a preliminary verification experiment on the whole model, producing a counterexample with an assignment to a quite large number of input variables. Indeed, the validation team, independent from the production team, has limited knowledge of the details of the equipment, and therefore was not able to interpret such a complex counterexample.
Actually, the experimental framework will allow us to perform a comprehensive study of the actual performance of Design Verifier over a set of control tables that implement different stations for which interlocking systems are being produced by the industrial partner. We will hopefully provide information about the actual effectiveness of performing automatic verification on large interlocking systems, giving indications about the optimal slice size that allow a meaningful verification within reasonable computation time for the single verification steps and minimal number of iterations.
However, the proposed iterative verification process, independently from the particular model checker used, has shown its capability of refining the knowledge of the validators on the internal working of a complex equipment, due to the incremental information given by the produced counterexamples. This is in our opinion the major contribution of this research.