TReX: A Tool for Reachability Analysis of Complex Systems

A. Annichini[1], A. Bouajjani[2], and M. Sighireanu[2]

[1] VERIMAG, Centre Equation, 2 av. de Vignate, 38610 Gières, France
[2] LIAFA, University of Paris 7, 2 place Jussieu, 75251 Paris Cedex 5, France

1 Introduction

Finite-state model-checkers such as SMV [13] and SPIN [11] do not allow to deal with important aspects that appear in modelling and analysing complex systems, e.g., communication protocols. Among these aspects: real-time constraints, manipulation of unbounded data structures like counters, communication through unbounded channels, parametric reasoning, etc.

The tool we propose, called TREX, allows to analyse automatically automata-based models equipped with variables of different kinds of *infinite*-domain data structures and with *parameters* (i.e., uninstantiated constants). These models are, at the present time, parametric (continuous-time) timed automata, extended with integer counters and communicating through unbounded lossy FIFO queues.

The techniques used in TREX are based on *symbolic reachability analysis*. Symbolic representation structures are used to represent infinite sets of configurations, and forward/backward exploration procedures are used to generate a symbolic reachability graph. The termination is not guaranteed, but efficient *extrapolation techniques* are used to help it. These techniques are based on computing the (exact) effect of the iteration of control loops detected dynamically during the search.

The kernel algorithm used in TREX is generic and can be used for any kind of data structures for which it is possible to provide a symbolic representation structure, a symbolic successor/predecessor function, and an extrapolation procedure. In the current version, TREX provides packages for symbolic representation of configurations of lossy FIFO channels and parametric timed automata and clock automata.

TREX allows to check on-the-fly safety properties, as well as to generate the set of reachable configurations and a finite symbolic graph. The set of reachable configurations can be used as an invariant of the system. For instance, if the analysed infinite-state model M is already an abstraction of a more concrete one M', the set of reachable configurations of M can be used to construct an invariant of M' which may help in its analysis. On the other hand, the generated

[3] This work was supported in part by the European Commission (FET project ADVANCE, contract No IST-1999-29082).

finite symbolic graph is a finite abstraction of the analysed model, which can be used for (conservative) finite-state model checking.

TReX is connected to the IF [5] environment which allows: (1) the use of high-level specification languages such as SDL, (2) the interaction with abstraction tools and invariant checkers such as InVeSt [3], (3) the use of finite-state model checkers such as CADP [8] and SPIN to verify properties on the finite symbolic graph.

TReX has been used to analyse several nontrivial protocols in their parametric versions, such as the Bounded Retransmission Protocol (BRP) [6]. This particular example requires the full power of TReX since it is a parametric heterogeneous model involving clocks, counters, and lossy channels. Moreover, the constraints manipulated in this model are nonlinear (contain products between variables). As far as we know, TReX is the only existing tool which allows to deal fully automatically with such a complex model. Indeed, tools like HyTech [10] and LPMC [12] deal with timed/hybrid automata and linear constraints, while LASH [15] deals with counter automata.

2 Architecture

Figure 1 shows the overall environment and architecture of TReX.

In addition to the description of the model in IF, the user of TReX can specify the initial constraints (invariants) on parameters, the initial symbolic configuration for the beginning of reachability analysis, and/or the safety property to be checked on-the-fly, expressed by an observer written in IF.

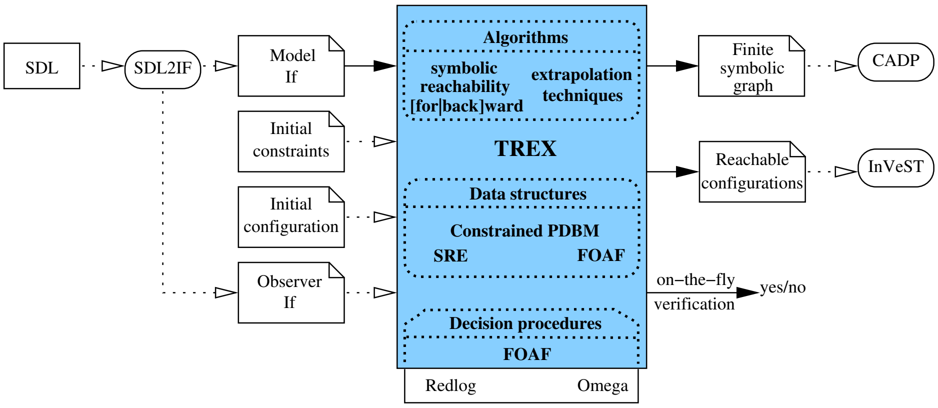

Fig. 1. Overview of the TReX's architecture and environment.

From the analysis of the input model, TReX instantiates *automatically* the generic reachability algorithm with the representation structures needed by the infinite data domains used.

Two such representations are actually provided in TReX. The first one is well suited for representing the contents of unbounded lossy FIFO-channels. We implemented a package for manipulating a class of regular expressions, *simple regular expressions* (SREs) [1], which is exactly the class of downward closed regular languages. This representation is interesting because the operations manipulating SREs during the symbolic analysis (the inclusion test, the effect of a transition, and the arbitrary number of executions of a control loop) are polynomial.

The second representation deals with sets of configurations of parametric timed automata and counter automata. We implemented a package for manipulating Constrained PDBMs (*Parametric Difference Bound Matrices*) [2]. The use of Constrained PDBMs allows to deal in a uniform way with counter/clock automata, parametric/non-parametric models, and systems generating linear or nonlinear arithmetical constraints. The package provides compact representation of PDBMs and efficient methods for operations used during symbolic analysis (e.g., emptiness check, intersection, and inclusion test). A special effort has been devoted to develop efficient representation of terms and formulas used in Constrained PDBMs, and simplification techniques on these objects. For this, we implemented a package, called FOAF (*First-Order Arithmetical Formulas*), which also gives the kind (linear or non-linear) of terms and formulas. This analysis is needed in order to apply the right decision procedure for the satisfiability of formulas.

The external decision procedures used actually by TReX are those offered by OMEGA [14] for formulas over integers and by the REDLOG package of REDUCE [9] for formulas over reals. Moreover, we implemented in the FOAF package the Fourier-Motzkin procedure [7] for elimination of quantifiers over real variables.

The symbolic graph generated by TReX is given by a couple of files: a file describing the transitions between reachable symbolic configuration given in the ALDEBARAN format and a file listing the reachable symbolic configurations. The ALDEBARAN file can be directly used for finite model-checking using the CADP tool. Reachable configurations may be used to extract new initial constraints (invariants) for the model and to do abstraction with InVeSt.

Each part of the TReX architecture has been implemented as an independent C++ module. This allows easy extension of TReX with new symbolic representations, analysis algorithms, and decision procedures.

3 Results and Future Work

TReX has been applied in a number of infinite state and/or parameterized protocols like: lift controller, Backery algorithm, BRP protocol, FDDI protocol, Fischer's protocol, alternating bit protocol (ABP), etc.

Table 1 gives the performances obtained by applying TReX on these examples. We consider two versions of TReX, depending on the package used for the decision procedure on reals: the first (*Standard*) uses the FOAF package and the second uses REDLOG.

The columns "*version*" specify the number of different kinds of variables used by each example: p for parameters, c for clocks, n for counters, $f(m)$ for lossy FIFO-channels with m messages, b for booleans, and $e(v)$ for enumerations with v values. The column "*# reach. conf.*" specifies the number of reachable symbolic configurations generated during symbolic analysis.

Table 1. Performance statistics on a Sun Ultra 10 (space in Mbytes, time in seconds).

Case study	*version*						*Standard*		with REDLOG		*# reach. config.*
	p	c	n	$f(m)$	b	$e(v)$	space	time	space	time	
Lift 10	-	-	3	-	-	-	6.5	7.52	6.5	7.52	8
Lift N	1	-	3	-	-	-	6.5	8.05	6.5	8.05	9
Backery	-	-	2	-	-	-	6.6	5.68	6.6	5.68	33
Fischer	2	2	-	-	-	1(3)	7	0.65	7	0.61	25
	2	3	-	-	-	1(4)	9.2	159.04	8.2	105.82	261
	2	4	-	-	-	1(5)	140	124 920	140	70 316	3 633
ABP	-	-	-	2(4)	-	-	6.9	0.05	6.9	0.05	8
FDDI	4	5	-	-	2	-	20	1 603.50	21	4 445	731
BRP	-	-	-	2(4)	-	-	6.8	0.30	6.8	0.30	36
	2	-	2	2(7)	4	-	16.4	195.93	16.4	195.93	173
	4	2	1	2(6)	2	-	89	5 518.57	85	5 563	106

The most complex example for which TREX has been applied is the BRP protocol. It is a timed file transfer protocol used by Philips. The three versions verified correspond to: (1) abstraction of clocks and counters—only lossy FIFO-channels are considered, (2) abstraction of clocks—counters and channels are used, (3) full version with channels, counters for the number of retransmissions, and clocks for timeouts. For the last version, TREX generates automatically the (non-linear) constraint needed to satisfy the timing response property of the protocol. The constraint relates three parameters of the protocol: the timeouts for the sender and for the receiver, and the number of retransmissions.

In future work, we plan to implement other data structures for the representation of configurations over counters and clocks, as well as to extend the input model to infinite nets of identical processes [4]. The version 1.0 of TREX is available at `http://www-verimag.imag.fr/~annichin/trex/`.

References

1. P.A. Abdulla, A. Bouajjani, and B. Jonsson. On-the-fly analysis of systems with unbounded, lossy, FIFO channels. In *Proceedings of the 10th CAV*, volume 1427 of *LNCS*, pages 305–317. Springer Verlag, 1998.

2. A. Annichini, E. Asarin, and A. Bouajjani. Symbolic techniques for parametric reasoning about counter and clock systems. In E.A. Emerson and A.P. Sistla, editors, *Proceedings of the 12th CAV*, volume 1855 of *LNCS*, pages 419–434. Springer Verlag, July 2000.
3. S. Bensalem, Y. Lakhnech, and S. Owre. InVeSt: A tool for the verification of invariants. In *Proceedings of the 10th CAV*, volume 1427 of *LNCS*. Springer Verlag, 1998.
4. A. Bouajjani, B. Jonsson, M. Nilsson, and T. Touili. Regular model checking. In E.A. Emerson and A.P. Sistla, editors, *Proceedings of the 12th CAV*, volume 1855 of *LNCS*, pages 403–418, July 2000.
5. M. Bozga, J.-C. Fernandez, L. Girvu, S. Graf, J.-P. Krimm, and L. Mounier. If: A validation environment for times asynchronous systems. In E.A. Emerson and A.P. Sistla, editors, *Proceedings of the 12th CAV*, volume 1855 of *LNCS*, pages 543–547. Springer Verlag, July 2000.
6. P.R. D'Argenio, J.-P. Katoen, T.C. Ruys, and J. Tretmans. The bounded retransmission protocol must be on time! In *Proceedings of 3rd Conference on Tools and Algorithms for the Construction and Analysis of Systems*, volume 1217 of *LNCS*, pages 416–432. Springer Verlag, 1997.
7. B.C. Eaves and U.G. Rothblum. Dines-fourier-motzkin quantifier elimination and an application of corresponding transfer principles over ordered fields. *Mathematical Programming*, 53:307–321, 1992.
8. J.-C. Fernandez, H. Garavel, A. Kerbrat, R. Mateescu, L. Mounier, and M. Sighireanu. Cadp (cæsar/aldebaran development package): A protocol validation and verification toolbox. In R. Alur and T.A. Henzinger, editors, *Proceedings of the 8th CAV*, volume 1102 of *LNCS*, pages 437–440. Springer Verlag, August 1996.
9. A.C. Hearn. *REDUCE — User's and Contributed Packages Manual*. Codemist Ltd., February 1999. version 3.7.
10. T.A. Henzinger, P.-H. Ho, and H. Wong-Toi. Hytech: A model checker for hybrid systems. *Software Tools for Technology Transfer*, 1(1):110–122, 1997.
11. G.J. Holzmann. *Design and Validation of Computer Protocols*. Software Series. Prentice Hall, 1991.
12. R.F. Lutje Spelberg, W.J. Toetenel, and M. Ammerlaan. Partion refinement in real-time model checking. In A.P. Ravn and H. Rischel, editors, *Proceedings of 5th FTRTFT*, volume 1486 of *LNCS*, pages 143–157. Springer Verlag, 1998.
13. K.L. McMillan. *The SMV system*. Cadence Berkeley Labs, 1999.
14. Omega Team. *The Omega Library*, November 1996. version 1.1.0.
15. P. Wolper and B. Boigelot. On the construction of automata from linear arithmetic constraints. In S. Graf and M. Schwartzbach, editors, *Proceedings of the 6th Conference on Tools and Algorithms for the Construction and Analysis of Systems*, volume 1785 of *LNCS*. Springer Verlag, 2000.