Translations for Model-Checking Temporal Logic with Past
(Extended abstract)

F. Laroussinie
LIFAC - ENS Cachan
61, av. Pdt Wilson
94235 Cachan Cedex
France
email: fl@lifac.ens-cachan.fr

Ph. Schnoebelen
Leibniz - IMAG
46, av. F. Viallet
38031 Grenoble Cedex
France
email: phs@imag.fr

Temporal logic model-checking [CG87] is a very promising approach to the formal verification of reactive systems. Today, this approach is followed by many research groups who continuously develop better, faster, more general methods. At the same time, the industrial community is becoming more and more aware of the potential benefits model-checking can bring.

Some strong points in favor of model-checking are

- it is fully automatic,
- it can efficiently handle complex systems,
- it yields more than just yes/no answers,
- temporal logic is a convenient language for reactive systems.

Still, the approach has some limitations that have been clearly identified. The main ones are:

- the state-explosion problem,
- the sometimes limited expressivity of CTL (+ fairness),
- the lack of modular methods,
- ...

These limitations have been attacked and some real progress has been made. Here we want to focus on the expressivity of CTL. CTL, the branching time Computation Tree Logic (essentially) introduced in [CES83, QS83] can be model-checked very efficiently: telling whether $M \models f$ can be done in time $O(|M| . |f|)$, which explains why CTL (with or without fairness) remains such a popular logic for model-checkers.

1 Extending CTL

CTL^*, B_μ (the branching-time modal calculus) and other logics have been suggested as possible extensions of CTL. The benefits are enhanced expressivity but the price is increased computational cost. E.g. model-checking CTL^* is PSPACE-complete, so that, in fact, no real-sized model-checker for these extended logics has been developped.

Another approach is to extend CTL with new constructs in such a way that the extended logic can still be translated back into CTL. In this sense, the new logic does not really extend CTL from a theoretical viewpoint. Still, it can be much of an improvement from a practical viewpoint. An early such proposal was the CTL^+ to CTL translation theorem [EH85] explaining how boolean combinators can be allowed between the linear-time modalities that must usually sit under the immediate scope of a path quantifier. Another, more recent, example (slightly outside the CTL limits) is [BG93].

In [LS95] we investigated how past-time constructs can be added to CTL without extending the theoretical expressive power. The main ingredients of our proposal were:

- X^{-1}, F^{-1}, S, the past-time equivalents of the usual X, F, U combinators,
- a semantics where past is *linear* (or determined), *finite*, and *cumulative*,
- a new combinator, N for “From Now On”, very useful in the few situations where a cumulative past is not the most convenient choice.

Our main result was a translation algorithm from $CTL + \mathsf{F} + \mathsf{N}$ into plain CTL and proofs that, in general, X^{-1} or S cannot be accomodated in the translation framework.

In this talk we consider a comprehensive example which can be seen as experimental evidence supporting our views that

- our semantic choice for the meaning of past is very-well suited to the specification of reactive systems,
- adding past-time really makes specifications clearer and simpler,
- the translation-based approach does not suffer, in practice, from theoretically possible combinatorial explosion problems.

As a bonus, the example suggests a precise discipline for using S and X^{-1}. The net result is a translation theorem for a CTL+Past fragment greatly extending the [LS95] result.

References

[BG93] O. Bernholtz and O. Grümberg. Buy one, get one free !!! In *Proc. 1st Int. Conf. Temporal Logic, Bonn, LNAI 827*, pages 210–224. Springer-Verlag, July 1993.

[CES83] E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications: A practical approach. In *Proc. 10th ACM Symp. Principles of Programming Languages, Austin, Texas*, pages 117–126, January 1983.

[CG87] E. M. Clarke and O. Grümberg. Research on automatic verification of finite-state concurrent systems. *Ann. Rev. Comput. Sci.*, 2:269–290, 1987.

[EH85] E. A. Emerson and J. Y. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. *Journal of Computer and System Sciences*, 30(1):1–24, 1985.

[LS95] F. Laroussinie and Ph. Schnoebelen. A hierarchy of temporal logics with past. *Theoretical Computer Science*, 148(2):303–324, 1995.

[QS83] J. P. Queille and J. Sifakis. Fairness and related properties in transition systems. A temporal logic to deal with fairness. *Acta Informatica*, 19:195–220, 1983.