A False History of True Concurrency: From Petri to Tools

Javier Esparza

Institut für Informatik, Technische Universität München
Boltzmannstr. 3, 85748 Garching, Germany

Carl Adam Petri passed away on July 2, 2010. I learnt about his death three days later, a few hours after finishing this text. He was a very profound and highly original thinker, and will be sadly missed. This note is dedicated to his memory.

This is an abstract of [1], a brief note describing the path of ideas that lead from the *theory of true concurrency*, a semantic theory about the nature of concurrent computation, to the *unfolding approach to model-checking*, a pragmatic technique for palliating the state-explosion problem in automatic verification. While the note provides a very incomplete and hence "false" view of true concurrency, it also includes several pages of references.

The theory of true concurrency started with two fundamental contributions by Carl Adam Petri in the 60s and 70s: *Petri nets*, the first mathematical formalism for asynchronous computation, and *nonsequential processes*, a semantics that proposes to order events not by means of global timestamps, but according to the *causality relation*. Then, in the early 80s Nielsen, Plotkin, and Winskel showed that, in the same way that the executions of a nondeterministic system can be bundled together into a *computation tree*, its nonsequential processes can be bundled together into the *unfolding* of the system, an object providing information about the causality and choice relations between events. The theory of unfoldings was further developed by Engelfriet, Nielsen, Rozenberg, Thiagarajan, Winskel, and others.

The goal of all this research was purely semantic: to mathematically define the behaviour of a concurrent system. During the 80s, model checking introduced a new (and very successful) view of the semantics of a system as an object that can be constructed and stored in a computer for verification purposes. However, model-checking concurrent systems faced the problem that the number of global states of a system may grow exponentially in the number of its sequential components. This *state-explosion* problem was attacked by McMillan in the early 90s in his PhD thesis. The thesis famously proposed the use of Binary Decision Diagrams, but in a different chapter it presents a second idea: instead of computing an initial part of the computation tree containing all global states (a *complete prefix*), McMillan suggested to construct a complete prefix *of the unfolding* containing the same information as the complete prefix of the computation tree, but encoded far more succinctly. This prefix becomes a data structure for compactly representing the set of global states of a concurrent system.

H. Ehrig et al. (Eds.): ICGT 2010, LNCS 6372, pp. 1–2, 2010.

McMillan's approach still faced two problems. First, while the complete prefix of the unfolding was usually much smaller than the complete prefix of the computation tree, it could also be *exponentially bigger* in the worst case. Second, the approach could at first only check specific properties, like deadlock freedom or conformance. Both problems were overcome during the 90s: improved algorithms for constructing complete prefixes were found, and extensions to (almost) arbitrary properties expressible in LTL were proposed.

Since 2000 the algorithms for constructing complete prefixes have been extended to many concurrency formalisms: a large variety of Petri net classes, communicating automata, process algebras, graph transformation systems, and others. They have also been parallelized and distributed. Unfolding-based verification techniques have been applied to conformance checking, analysis and synthesis of asynchronous circuits, monitoring and diagnose of discrete event systems, analysis of asynchronous communication protocols, and other problems. Today, the unfolding approach is a good example of how abstract, speculative considerations about the nature of computation can evolve into pragmatic techniques for automatic verification.

Reference

1. Esparza, J.: A False History of True Concurrency: from Petri to Tools. In: Weber, M. (ed.) SPIN 2010. LNCS, vol. 6349, pp. 180–186. Springer, Heidelberg (2010)