Layered reasoning for randomized distributed algorithms

Mani Swaminathan1, Joost-Pieter Katoen2 and Ernst-Rüdiger Olderog1

1 University of Oldenburg, Oldenburg, Germany. E-mail: mani.swaminathan@informatik.uni-oldenburg.de
2 RWTH Aachen University, Aachen, Germany

Abstract. This paper adopts the communication closed layer (CCL) concept of Elrad and Francez to the formal reasoning of randomized distributed algorithms. We do so by enriching probabilistic automata (PA) with a layered composition operator, an intermediate between parallel and sequential composition. Layered composition is used to establish probabilistic counterparts of the CCL laws that exploit independence and/or precedence conditions between the constituent PA. The probabilistic CCL laws enable partial order (po-) equivalence when layered composition is replaced by sequential composition. Such po-equivalence induces a purely syntactic partial-order state space reduction via layered separation in compositions of PA while preserving probabilistic next-free linear-time properties. The feasibility of such layered separation is demonstrated on a randomized mutual exclusion algorithm by Kushilevitz and Rabin, complementing an algebraic approach (for analyzing this algorithm) by McIver, Gonzalia, Cohen, and Morgan.

Keywords: Probabilistic automata, Layered composition and separation, Communication closedness, Partial order equivalence, Randomized mutual exclusion

1. Introduction

Randomized distributed algorithms are intricate. Randomization is of paramount importance in the setting of distributed algorithms. It is used to break the symmetry between identical process components in leader election and mutual exclusion, for routing purposes, or for obtaining consensus—a problem that is known to be unsolvable in a deterministic setting. The design and analysis of randomized distributed algorithms is however highly non-trivial. Lehmann and Rabin, e.g., argue that “proofs of correctness for probabilistic distributed systems are extremely slippery” [LR81], and various flawed versions of randomized distributed algorithms exist, cf. [Seg00]. This is mainly due to the fact that the stochastic process describing the evolution of a randomized distributed algorithm changes depending on the generally unknown scheduling policies and relative speeds of the individual process components, entailing a complex interplay between randomization and nondeterminism.
Why probabilistic automata? Probabilistic automata [SL95, Seg00] (PA) constitute an operational framework for the modelling and analysis of discrete systems that exhibit both nondeterministic and randomized behavior, such as randomized distributed algorithms. An I/O-variant of PA (PIOA) has been used to successfully analyze intricate randomized distributed algorithms such as the Aspnes–Herlihy randomized consensus algorithm [PSL00] and the IEEE Firewire protocol [SV99]. Extensions of PIOA have been used for specifying and verifying security protocols [CCK’08]. In the context of concurrency theory, PA are used as semantic models for stochastic process algebras, and have been equipped with (bi)simulation notions [SL95].

Layering. Despite the presence of modular verification techniques for PA [Seg00], the correctness proofs of randomized distributed algorithms remain difficult and require substantial human ingenuity. This paper attempts to simplify their reasoning by enriching probabilistic automata with the concept of layering. The main underlying idea is that the computations of randomized distributed algorithms often exhibit a sequential (i.e., layered) structure. The idea of using such sequential structure to simplify the verification of distributed algorithms was originally proposed in the eighties by Elrad and Francez [EF82], and has been extended, formalized [SdR94], and applied to intricate distributed algorithms such as the minimal spanning tree algorithm [JZ92] about a decade later. Layered reasoning (though in a different way as in this paper) has been recently used to obtain tighter bounds for asynchronous randomized consensus [AC08]; earlier work on applying layering to bound analysis appeared in [MR02]. Most recently, a notion of layering for distributed real-time systems (modelled as networks of timed automata) has been investigated in [OS10]. To our knowledge, layering has not been applied to ease the verification of randomized distributed algorithms. We therefore study in this paper layered reasoning for randomized distributed algorithms, with PA (enriched with shared data variables) as the underlying operational model. Our layered reasoning here builds on that investigated in [OS10] for the non-randomized, real-time setting.

Our contributions. For simplifying the formal reasoning of randomized distributed algorithms, we introduce layered composition $P \parallel Q$ of PA $P$ and $Q$: the PA $P \parallel Q$ behaves like $P || Q$, the parallel composition of $P$ and $Q$, except that for all actions $a$ of $P$ and $b$ of $Q$ that depend on each other (e.g., as both actions affect the same shared variable), $a$ is executed before $b$. Layered composition is thus a kind of asymmetric parallel composition. We obtain a probabilistic version of the communication closed layer (CCL) law that allows us to identify $(P_1 \parallel P_2) || (Q_1 \parallel Q_2)$ and $(P_1 || Q_1) \bullet (P_2 || Q_2)$, provided $P_1, Q_2$ and $P_2, Q_1$ respect pairwise certain independence or precedence conditions. This CCL law enables us to transform a randomized distributed algorithm into an equivalent layered one so as to permit easier verification. This verification is technically enabled using a partial-order (po) equivalence on probabilistic automata; in particular we show that $P \parallel Q$ and $P; Q$ are po-equivalent. The notion of po-equivalence is shown to preserve probabilistic next-free linear temporal logic properties. The CCL law together with the po-equivalence of $\parallel$ and $;$ allows the transformation of $(P_1 || P_2) || (Q_1 || Q_2)$—via intermediate representations $(P_1 \parallel P_2) || (Q_1, Q_2)$ and $(P_1 || Q_1) \bullet (P_2 || Q_2)$—finally to $(P_1 || P_2) || (Q_1 || Q_2)$ under the aforementioned independence or precedence conditions. This yields a syntactic (partial order) state-space reduction that may be applied prior to automated model checking of randomized distributed algorithms (as in [KN02]). We illustrate the feasibility of probabilistic layering on the (non-trivial) randomized mutual exclusion algorithm of Kushilevitz and Rabin [KR92]. This algorithm improves an earlier version proposed by Rabin in [Rab82], whose correctness proof was demonstrated as being flawed in [Sat92], due to the comparison of two probabilities that were not defined within the same probability space [Seg00].

Relation to the work of Carroll Morgan. Since the mid-nineties, the research of Carroll Morgan has largely been centered on the (algebraic) specification and verification of probabilistic programs described in a probabilistic guarded command language (pGCL). This research has been consolidated in the monograph co-authored with Annabelle McIver [MM04]. More recently, Morgan (along with his co-authors McIver, Gonzalia, and Cohen) advanced the pGCL-based algebraic specification and verification approach by a probabilistic Kleene Algebra (pKA) [MGCM08], with application to randomized distributed algorithms. This pGCL-/pKA-based approach compares to our layering as follows:

- Central to the pGCL/pKA approach in [MGCM08] is the exploitation of the separation theorems introduced earlier in [Coh00] for the non-randomized setting. Such separation theorems simplify the (algebraic) reasoning of (randomized) distributed systems by reducing complex interleavings into “separated” behaviours that admit individual analysis [MGCM08]. Our layering principle is similar in spirit: we exploit structural