Model And Verify CPS Using "Probabilistic Automata"

Syed Muhammad Abis Rizvi[1]

Contents

Abstract: A simple introduction to the probabilistic automata (PA) model is given in this paper. Probabilistic automata is a modeling formalism that can be used to formally describe both nondeterministic and probabilistic aspects of real-time systems . This variation of probabilistic automata is extended with discrete probability distributions. It also allows for model checking against probabilistic timed temporal logic properties.Wireless communication protocols, automobile network protocols, and randomized security protocols are examples of applications. This paper gives an introduction to Probabilistic Automata and describes techniques used for analysing and verifying cps systems.

[1] syed-muhammad-abis.rizvi@stud.hshl.de

1 Motivation

Model checking is one of the many automated verification approaches that offer effective ways to analyze systems correctness with precision. This research must increasingly consider quantitative elements of the systems under evaluation, such as real-time properties and probabilistic behavior. Michael O. Rabin introduced this concept in 1963 and was earlier known as Rabin automaton.The probabilistic automata (PA), which is a generalization of the nondeterministic finite automaton in mathematics and computer science, incorporates the probability of a given transition into the transition function, converting it into a transition matrix. Thus, the concepts of a Markov chain and a subshift of finite type are also generalized by the probabilistic automaton. Stochastic languages, of which regular languages are a subset, are those understood by probabilistic automata. There are an infinite number of stochastic languages. A probabilistic automata (PA) has a weighted set (or vector) of next states, where the weights must add up to 1, and are hence probabilities (making it a stochastic vector). To incorporate these weights, it is also necessary to change the conceptions of states and acceptance. A stochastic vector of states must now additionally describe the machine's state at a particular step, with a state being accepted if its overall probability of being in an acceptance state exceeds a set threshold.These systems include probabilistic communication protocols like the IEEE 1394.Root Contention protocol [IEE96] and the Binary Exponential Back Off protocol, see [Tan81], randomized, distributed algorithms like the randomized dining philosophers, and fault tolerant systems(such as unreliable communication channels.[St02].State transition systems serve as the foundation for PAs, later we will distinguish between both probabilistic and nondeterministic choices. Further we will discuss basic similarities and differences between both types.

2 Probablistic Automata

An ordinary automaton (also known as a labeled transition system or state machine) is the same as a probabilistic automata with the exception that the goal of a transition is a probabilistic selection from among numerous future states. We quickly review the idea of a nonprobabilistic automaton, often referred to as a state machine, a labeled transition system, or a state transition system, before delving into the specifics of this. Where as nonprobabilistic automata or NA, is a structure made up of states and transitions, which are sometimes known as steps. The states of a NA correspond to the modeled system's states. As the system's starting configuration, one or more states are marked as start states. The NA's transitions are marked by actions and signify modifications to the system states. As previously stated, the main distinction between a probabilistic automaton and a nonprobabilistic automaton is that the latter's objective of a transition is now a probabilistic option among numerous next states rather than a single one.

The set of a PA's trace distributions as its semantics. Each trace distribution is a space of probabilities that gives particular sets of traces a probability. According to the theory, a trace

distribution results from first answering the nondeterministic decision and then abstracting from nonvisible, that is, by eliminating the states and internal actions and leaving only the exterior ones. The theory of PAs permits nondeterministic decisions to be resolved by probabilistic decisions, which is a distinction from the nonprobabilistic situation. We will see that an adversary describes these. Additionally, unlike in a NA, a PA's resolution of nondeterministic choices results in a tree-like structure rather than a linear execution. We start by reviewing the meaning of traces for NAs.

3 Markov decision processes and stochastic games

MDPs, or Markov decision processes, are a popular formalism for simulating systems with probabilistic and nondeterministic behavior.

4 Timed Automata / Timed

Similar to the NA model, timing can be incorporated into the PA model (see öld fashioned formula for time"[AL92]). A PA with time passage actions is known as a probabilistic timed automaton (PTA). These events show that d time units have passed (d 2 R>0). While time passes, nothing else happens, and according to the PTA method, time moves deterministically. Therefore, none of the time passage acts, in particular, may specify (internally) nondeterministic or probabilistic options; for details, see the definition's criteria 1 and 2. The PTA's state must be clearly specified at each moment in time for the third of the conditions below, known as Wang's Axiom [Yi90], to hold true. Additionally, two consecutive time passage activities must be capable of being performed simultaneously. The third need below, known as Wang's Axiom [Yi90], calls for the PTA's state to be clearly specified at each point in time as time progresses and, conversely, for two successive time passage actions to be able to be integrated into a single action.

4.1 Path and Traces

The set of a NA's traces defines its semantics. The following procedures are used to obtain a trace, which depicts one of the system's possible visible behaviors. The NA's nondeterministic choices are first resolved. This results in an execution, or a series of states and deeds. The execution is then freed from the states and internal operations. This results in a trace—a series of actions.

4.2 Trace Distribution

The set of distributions over a PA's traces, often known as the "trace distributions,"define the semantics of the PA. Similar to how a trace in a NA indicates one of the system's possible

apparent behaviors, each trace distribution of a PA does the same. As previously stated, in order to produce a trace distribution, the nondeterministic choices in the PA must first be resolved (replaced by probabilistic choices), and only then can the states and internal actions be eliminated. Formally, partial, randomized adversaries are used to resolve nondeterminism. In a NA, an adversary could be compared to an execution. As a result, every adversary results in a unique system behavior. Each adversary is linked to a probability space that assigns probabilities to certain sets of pathways in order to investigate the probabilistic behavior it generates. After that, the related probability space is purge of all states and internal activities to produce the trace distribution of this opponent.

To model systems with probabilistic choice, many models have been put out in the literature. These models can be categorized based on the kinds of probabilistic and nondeterministic options they support. We take into account models with either no, simply external, or both internal and external nondeterminism. In terms of probability, we take into account models with discrete, exponential, and other types of probabilities.

	none	external	ful
discrete	DTMC	MDP	pr A
exponential	CTMC SPN	PIOA	a
general	SMP GSNP	SMDP	

Fig. 1: Caption

5 Conclusion

Discrete probabilistic choice and nondeterministic choice are combined orthogonally in the probabilistic automaton model addressed in this study. This makes the model appropriate for reasoning about randomized distributed algorithms, probabilistic communication protocols, and systems with failing components. It also enables us to develop an asynchronous parallel composition operator. Nonprobabilistic transition systems, Markov chains, and Markov decision processes are all included in PAs.

6 Declaration of Originality

I, ..., herewith declare that I have composed the present paper and work by myself and without the use of any other than the cited sources and aids. Sentences or parts of sentences quoted literally are marked as such; other references with regard to the statement and scope are indicated by full details of the publications concerned. The paper and work in the same or similar form have not been submitted to any examination body and have not been published. This paper was not yet, even in part, used in another examination or as a course performance. I agree that my work may be checked by a plagiarism checker.

Date&Place - Full Name

Bibliography

[St02] Stoelinga, Mariëlle: An introduction to probabilistic automata. Bulletin of the EATCS, 78(176-198):2, 2002.

[AL92] M. Abadi and L. Lamport. An old-fashioned recipe for real time. In J.W. de Bakker, C. Huizing, W.P. de Roever, and G. Rozenberg, editors, Proceedings REX Workshop on Real-Time: Theory in Practice, Mook, The Netherlands, June 1991, volume 600 of Lecture Notes in Computer Science, pages 1–27. Springer-Verlag, 1992.

[IEE96] IEEE Computer Society. IEEE Standard for a High Performance Serial Bus. Std 1394-1995, August 1996.

[Tan81] A.S. Tanenbaum. Computer networks. Prentice-Hall International, Englewood Cliffs, 1981.

[Yi90] Wang Yi. Real-time behaviour of asynchronous agents. In J.C.M. Baeten and J.W. Klop, editors, Proceedings CONCUR 90, Amsterdam, volume 458 of Lecture Notes in Computer Science, pages 502–520. Springer-Verlag, 1990