Probabilistic Distributed Temporal Logic

Luís Morão, Jaime Ramos (Adviser)

Janeiro, 2011

Abstract. We present a logic for reasoning about temporal properties of distributed systems and finitely additive probability. The language for this logic allows us to make statements such as "The probability of 'sometime in the future φ' is at least s".
We start by presenting a linear temporal logic LTL with a sound and complete axiomatic system. We next present a distributed temporal logic DTL and show that this logic is reducible to LTL.
To introduce probabilities in the linear temporal logic we define pLTL. We then make a first attempt for probabilities in a distributed temporal logic with $\mathsf{PDTL}_{\mathsf{L}}$ by having probabilities in the temporal states. Finally we adopt for an exogenous approach, $\mathsf{PDTL}_{\mathsf{G}}$, with probabilities over the models of a distributed temporal logic.

Extended Abstract

The term temporal logic is used to describe any system of rules and symbolism for representing, and reasoning about, propositions qualified in terms of time. It was first introduced as tense logic, by Artur Prior in the 1960s , as a particular modal logic-based system [33].

Temporal logic is a branch of logic focusing on propositions whose truth values depends on time. It has been studied and developed by practitioners mainly in the areas of computer science and logic. One main concern is with the study of *distributed systems*, which are collections of connected autonomous components. The first advances in the use of temporal logics to study distributed systems were made by Pnueli, followed by Sernadas [35, 31]. This logic was also used to study *reactive systems*, which are system that maintain ongoing processes, and to study any form of *concurrency* which is the property of systems running several processes at the same time.

Temporal logic provides a formalism for describing the occurrence of events in time that is suitable for reasoning about concurrent programs [23]. When

a program runs two processes in parallel we can't infer the input-output relation based solely on the input-output relations computed by each of the individual processes. The reason for this is that, running in parallel, the processes may interfere with one another, resulting in a different behavior than if they run alone. Temporal logic was then used to describe the desired behavior or operation of a system, while avoiding references to the method or details of it's implementation. In order to analyze systems, Manna and Pnueli used a simplified model in which the executions were restricted to be interleaved [23]. An interleaved execution is one in which at any instant only one process is executing an instruction. Once the instruction was completed, another process would initiate an instruction. Under this model, the execution of the program proceeds as a sequence of discrete steps. The selection of the next process to be executed was made by a *scheduler*. This however raised a problem of *fairness*. An infinite process could be constantly selected by the scheduler, which could be neglecting another process that was ready to execute an instruction. A fair scheduler would ensure that no process would be neglected forever.

Hence the behavior of a concurrent program is characterized by the set of its fair execution sequences. Temporal logic is then utilized to state properties of the execution sequences of a given program, thus describing properties of the dynamic behavior of the program.

Another task of great importance is the verification of a program, proving or disproving the correctness of the algorithms. The old approach was to construct a proof by hand using axioms and inference rules in a deductive system such as temporal logic [23]. These proofs took some effort and brilliance to organize and, due to the complexity of testing validity, mechanical theorem provers were of little help. Clarke and Emmerson presented a model-theoretic approach for the case of finite-state concurrent systems that would mechanically determine if the system met a specification expressed in propositional temporal logic [9]. This technique is known as *model checking*.

They used a specification language they called computation tree logic (CTL) that is a proposition, branching-time temporal logic. The global state graph of the concurrent system was viewed as a finite Kripke structure, and an efficient algorithm could be given to determine whether the structure was a model of a particular formula (therefore determining if the program met its specification).

However, they encountered the problem that with CTL it was not possible to assert that correctness only holds on fair execution sequences [3, 10]. They overcame this problem by moving the fairness requirements into the semantics of CTL.

When defining temporal logic, there are two possible views regarding the underlying nature of time. One is that time is linear: At each moment there is only one possible future. The other is that time has a branching, treelike nature: At each moment, time may split into alternate courses representing different possible futures [3].

The modalities of a temporal logic system usually reflect the semantics regarding the nature of time. Thus, in a logic of linear time, temporal modalities are provided for describing events along a single time path. In contrast, in a logic of branching time, the modalities reflect the branching nature of time, by allowing quantification over possible futures.

For verification purposes we are typically interested in properties that hold for all computation paths. It is thus satisfactory to pick an arbitrary path and reason about it, asserting the existence of alternative computation paths as provided by a branching time logic like CTL, and so it serves a purpose as model checking.

We are however more concerned with results that will help to verify correctness of concurrent programs, for which a linear time logic like LTL is generally used. This is the reason why LTL is taken as our base logic.

As a side note, there is also the language CTL* that merges linear time with branching time, allowing linear time assertions prefixed by existential path quantifiers [3].

The use of branching temporal logics like CTL and CTL* and even linear temporal logics as been deeply studied for reasoning about concurrent systems [22, 31]. However, there was a need for a precise notion of causality as reflected by the time structures adopted for logics. This led to the study of partial order temporal logics [32].

The idea of a partial order temporal logic is that it adopts a local causal perspective, instead of taking a look at the entire system. This is the case of the logics of partial ordered computations introduced by Printer and Wolper , or of interleaving set temporal logics and of temporal logics for reasoning about distributed transition systems, with product state spaces, occurrence nets and event structures [32].

In particular, n-agent temporal logics are in the latter category. Event structures are enriched with information about its sequential agents as models of concurrent systems.

In this approach each individual agent has a only a partial view of the whole distributed system in a particular time instant, due to the autonomy of each agent, and therefore communication has an important role to link up the system. n-agent logics can explicitly distinguish sequential agents in the system, refer to the local viewpoint of each agent and express communication between agents.

There are several versions of n-agent logics, with different perspectives on how non-local information can be accessed by each agent. Caleiro and Ehrich proposed a logic which assumes that an assertion about another agent is only possible at a communication point [11]. For example some logics assume that at each instant, the actual information about another agent is the one corresponding to the last communication with it.

Other n-agent logics consider the existence of a "present-knowledge" allowing reference to non-local properties of agents.

We follow the approach of Caleiro and Ehlrich when dealing with concurrent systems [11, 5].

Recently there's been an increasing interest in probabilistic logic due to the growing importance of probability in security and in quantum logic. Adding probability features to a base logic has been a research topic for some time [2, 15, 37]. One of the first attempting to combine logic and probability was Carnap [7].

Several approaches to enriching the base languages introduce a probability operator which allows the construction of new formulas and terms from the base formulas e.g. [1, 34]. From a semantic point of view there are two basic approaches. Either the models of the base logic are modified so that they accommodate the probability component (endogenous approach) or the models are kept as they are and the probabilities are assigned outside the models (exogenous approach).

The exogenous approach to enriching logics was introduced by Mateus and Sernadas while developing a new approach for quantum logic [26]. However it is shown that it can be used to enrich any logic [25].

The idea is to add probabilities features to a logic without changing it's base structure. The probability language is defined by taking the base formulas as terms and having a probability operator as a term constructor. This approach will be most convenient when we have to deal with languages that have some delicate formulas, as is the case of formulas for communication between agents.

When using probabilities, there seems to be two kind of statements that we may want to express. One is what Hacking calls the *chance setup* , that is a statement about what one might expect as the result of performing some experiment in a given situation, e.g. " there's s probability of randomly choosing an agent for which formula φ is true" [14].

The other statement captures what has been called a *degree of belief* [4]. These statements are about quantifying an unknown property of a situation, e.g. "there's s probability that formula φ of agent A is true ". This refers to a particular agent, and expresses our belief in whether agent A has the said attribute.

The difference between the two statements is that the first statement assumes a fixed universe, and makes an assertion about that universe. The second statement implicitly assumes the existence of a number of possibilities with some probability over these possibilities [16]. The Statements that concerns us are of the second type, as we want to represent a system's behavior, allowing all the possible realities.

The first statement is connected to the endogenous approach, where the probability is assigned to formulas inside the logic [29, 12]. The second statement is closer to an exogenous concept, which requires that we define a higher-level logic which will be the probability logic. A model for this logic consists of a probability measure on the space of possible valuations of the base logic.

In this paper we take several steps to define an exogenous probabilistic logic for reasoning about concurrent systems. We start by laying the foundations and defining a language for a linear temporal logic LTL with a sound and weakly complete axiomatic system, following the works [19, 35, 22, 31]. Next we follow with the definition of a distributed temporal logic DTL, based on the works of [23, 11, 8, 6, 5]. This will be an extension of the previous defined logic LTL to include additional time lines for the components that constitute a system. An axiomatic system is also presented, but we don't show results

for completeness, although results of this nature have been shown in [5] for a different deductive system for DTL. Not only that but we will also show that there's a reduction from DTL to LTL, that is, if we are dealing with a system of only one agent, it can be represented by LTL.

In the second half of this work we deal with enriching the base logics LTL and DTL for probabilistic reasoning. For this, we start by defining a linear temporal logic with a probability operator pLTL, that only has probability on classical propositional formulas. Temporal formulas, and *actions*, which are multi-valued symbols, won't have a probability assigned to them. We present an axiomatic system, following mainly the work [30]. We also prove some results for formula manipulation. Lastly, we show that if all classical formulas have probability 1, then pLTL can be reduced to LTL.

To enrich DTL with probabilities, we take two steps. The first step is similar to what we done with pLTL, defining a probability operator only for classical propositional formulas. Thus we define a logic pDTL_L that defines probabilities at a local level, an exogenous approach that considers a probability measure for each time event.

The second step for a probabilistic DTL takes a different approach of what has been done with pLTL and pDTL_L. Instead of a local operator for classical formulas, we follow the works of [24, 8, 27, 25] with an exogenous approach that will include all formulas. Probabilities are defined at a global level over the models of DTL, resulting in the logic pDTL_G. As a consequence of the exogenous approach, the base structure of DTL is kept intact, allowing probabilities on all formulas, which include classical and temporal formulas, as well as communication and actions. The results we present in this section are mainly about probabilistic reasoning and specific manipulation of formulas, with some focus on actions and communication.

References

[1] M. Abadi and J.Y. Halpern, Decidability and expressiveness for first-order logics of probability. *Inform. Comput. 112 1* (1994), pp. 1–36.

[2] E. Adams. *A primer of probability Logic*. Cambridge University Press, Cambridge, England, 1997.

[3] E. Allen Emerson , Joseph Y. Halpern, "Sometimes" and "not never" revisited: on branching versus linear time temporal logic, *Journal of the ACM (JACM)*, v.33 n.1, p.151-178, Jan. 1986 .

[4] F. Bacchus. *Representing and Reasoning with probabilistic Knowledge*. MIT Press, 1990.

[5] D. Basin, C. Caleiro, J. Ramos, and L. Viganò. Labelled tableaux for distributed temporal logic. *Journal of Logic and Computation*, 19:1245–1279, 2009.

[6] D. Basin, C. Caleiro, J. Ramos, and L. Viganò. Distributed temporal logic for the analysis of security protocol models. Preprint, SQIG - IT and IST - TU Lisbon, 1049-001 Lisboa, Portugal, 2010. Submitted for publication..

[7] R. Carnap. *Logical Foundations of Probability.* The university of Chicago Press, 1950.

[8] R. Chadha , L. Cruz-Filipe , P. Mateus , A. Sernadas, Reasoning about probabilistic sequential programs, *Theoretical Computer Science,* v.379 n.1-2, p.142-165, June, 2007.

[9] E. M. Clarke , E. A. Emerson , A. P. Sistla, Automatic verification of finite state concurrent system using temporal logic specifications: a practical approach, *Proceedings of the 10th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, p.117-126, January 24-26, 1983, Austin, Texas

[10] E. Clarke and E. Emmerson. Characterizing Properties of Parallel Programs as Fixedpoint, In Seventh International *Colloquium on Automata, Languages, and Programming,* Volume 85 of LNCS, 1981.

[11] H.-D. Ehrich and C. Caleiro. Specifying communication in distributed information systems. *Acta Informatica,* 36:591–616, 2000.

[12] R. Fagin, J. Halpern and N. Megiddo. A logic for reasoning about probabilities, *Information and Computation* 87:1,2, 1990, pp. 78-128.

[13] D. Gabbay, A. Pnueli, S. Shelah, J. Stavi, On the temporal analysis of fairness, in: *Proceedings of the 7th ACM Symposium on Principles of Programming Languages*, 1980, pp. 163–173.

[14] I. Hacking. *Logic of Statistical Inference.* Cambridge University Press; Toronto: Macmillan of Canada, 1965.

[15] A. Hájek. Probability, Logic, and Probability Logic, in Goble, Lou, ed., *The Blackwell Guide to Philosophical Logic*, Blackwell, 2001.

[16] J.Y. Halpern, An analysis of first-order logics of probability. *Arti. Intell.* 46 (1990), pp. 311– 350.

[17] Joseph Y. Halpern , Ron vander Meyden , Moshe Y. Vardi, Complete Axiomatizations for Reasoning about Knowledge and Time, *SIAM Journal on Computing*, v.33 n.3, p.674-703, 2004

[18] S. Hart , M. Sharir, Probabilistic temporal logics for finite and bounded models, *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, p.1-13, December 1984.

[19] F. Kröger, S. Merz. *Temporal Logic and State Systems*, Springer, 2008

[20] O. Lichtenstein, A. Pnueli, and L. Zuck. The Glory of the Past. In *Proceedings of the Conference on Logics of Programs*, volume 193 of Lecture Notes in Computer Science, pages 196-218. Springer-Verlag, 1985.

[21] O. Lichtenstein and A Pneuli. Propositional temporal logics: decidability and completeness, *Logic Jnl IGPL* 2000.

[22] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems.* Springer-Verlag, New York, 1992

[23] Z. Manna, and A. Pnueli. bf Verification of Concurrent Programs: The Temporal Framework. In R.S. Boyer and J.S. Moore, editors, *The Correctness Problem in Computer Science*, pages 215-273. Academic Press, London, 1981.

[24] P. Mateus, A. Pacheco, and J. Pinto. Observations and the probabilistic situation calculus. In D. Fensel, F. Giunchiglia, D. McGuinness, and M.-A. Williams, editors, *Procs. Eighth International Conference on Principles of Knowledge Representation and Reasoning* (KR2002), pages 327–338. Morgan Kaufmann, 2002.

[25] P. Mateus, A. Sernadas, and C. Sernadas. Exogenous semantics approach to enriching logics. In G. Sica, editor, *Essays on the Foundations of Mathematics and Logic*, volume 1, pages 165–194. Polimetrica, 2005.

[26] P. Mateus and A. Sernadas. Exogenous quantum logic. In W. A. Carnielli, F. M. Dionísio, and P. Mateus, editors, *Proceedings of CombLog'04, Workshop on Combination of Logics: Theory and Applications*, pages 141–149, 1049-001 Lisboa, Portugal, 2004. Departamento de Matemática, Instituto Superior Técnico.

[27] P. Mateus and A. Sernadas. Weakly complete axiomatization of exogenous quantum propositional logic. *Information and Computation*, 204(5):771–794, 2006. ArXiv math.LO/0503453.

[28] Z. Manna, and A. Pnueli. bf Verification of Concurrent Programs: A Temporal Proof System. In J.W. de Bakker and J. Van Leeuwen, editors, *Foundations of Computer Science IV, Distributed Systems: Part 2*, pages 163-255. Mathematical Centre Tracts 159, Center for Mathematics and Computer Science (CWI), Amsterdam, 1983.

[29] N. Nilsson. Probabilistic Logic, *Artificial Intelligence*, 28(1):71-87, 1986. .

[30] Z. Ognjanović, *Discrete Linear-time Probabilistic Logics: Completeness, Decidability and Complexity*, Vol.16 No.2, 2006. Oxford University Press.

[31] A. Pnueli. The Temporal Logic of Programs. In *Proceedings of the 18th IEEE Symposium Foundations of Computer Science* (FOCS 1977), 1977.

[32] S. Pinter and P. Wolper, A temporal logic for reasoning about partially ordered computations. In: *Proc. 3rd ACM Symposium on Principles of Distributed Computing*, 1984, pp. 28–37..

[33] A. Prior. *Time and Modality.* Oxford: Oxford University Press, 1958.

[34] K. Segerberg. Qualitative probability in a modal setting. In: *Proe. of the Second Scandinavian Logic Symposium* (ed. JE Fenstad), North Holland. 1971.

[35] A. Sernadas. Temporal aspects of logical procedure definition. *Information Systems*, 5(3):167–197, 1980.

[36] A. P. Sistla , E. M. Clarke, The complexity of propositional linear temporal logics, *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, p.159-168, May 05-07, 1982, San Francisco, California, United States

[37] J. Williamson. in Dov Gabbay, Ralph Johnson, Hans Jurgen Ohlbach & John Woods (eds)[2002]: Handbook of the Logic of Inference and Argument: The Turn Toward the Practical, *Studies in Logic and Practical Reasoning Volume 1*, Elsevier, pp. 397-424.