Behavioral QLTL

Giuseppe Perelli [*]
Sapienza University of Rome

Since the very early time of AI, researchers have tried to reduce planning to logical reasoning, i.e., satisfiability, validity, logical implication (Green 1969). However as we consider more and more sophisticated forms of planning this becomes more and more challenging, because the logical reasoning we need to do is intrinsically second-order. One prominent case is if we want to express the model of the world (aka the environment) and the goal of the agent directly in Linear-time Temporal Logic, which is the logic used most in formal method to specify dynamic systems. Examples are the pioneering work on using temporal logic as a sort of programming language through the MetateM framework (Barringer et al. 1995), the work on temporal extended goals and declarative control constraints (Bacchus and Kabanza 1998; Bacchus and Kabanza 2000), the work on planning via model-checking (Cimatti et al. 1997; Cimatti and Roveri 2000; Daniele, Traverso, and Vardi 1999; Bertoli, Cimatti, and Roveri 2001), the work on adopting LTL logical reasoning (plus some meta-theoretic manipulation) for certain forms of planning (Cerrito and Mayer 1998; Calvanese, De Giacomo, and Vardi 2002). More recently the connection between planning in nondeterministic domains and (reactive) synthesis (Pnueli and Rosner 1989) has been investigated, and in fact it has been shown that planning in nondeterministic domains can be seen in general terms as a form of synthesis in presence of a model of the environment (Camacho, Bienvenu, and McIlraith 2019; Aminof et al. 2019), also related to synthesis under assumptions (Chatterjee and Henzinger 2007; Chatterjee, Henzinger, and Jobstmann 2008).

However the connection between planning and synthesis also clarifies formally that we cannot use directly the standard forms of reasoning in LTL, such as satisfiability, validity, or logical implication, to do planning. Indeed the logical reasoning task we have to adopt is a nonstandard one, called "*realizability*" (Church 1963; Pnueli and Rosner 1989), which is in inherently a second-order form of reasoning on LTL specifications. So one question comes natural: can we use the second-order version of LTL, called QLTL (or QPTL) (Sistla 1985) and then avoid use nonstandard form of reasoning?

In (Calvanese, De Giacomo, and Vardi 2002) a positive answer was given limited to conformant planning, in which we cannot observe response of the environment to the agent actions. Indeed it was shown that conformant planning could be captured through standard logical reasoning in QLTL. But the results there do not extend to conditional planning (with or without full observability) in nondeterministic environment models. The reason for this is very profound. Any plan must be a "*process*", i.e., observe what has happened so far (the history), observe the current state and take a decision on the next action to do (Abadi, Lamport, and Wolper 1989). QLTL instead interprets quantified propositions (i.e., in the case of planning, the actions to be chosen) through functions that have access to the whole traces, i.e., also the future instants, hence they cannot be considered processes. This is a clear mismatch that makes standard QLTL unsuitable to capture planning through standard reasoning tasks.

This mismatch is not only a characteristic of QLTL, but, interestingly, even of logics that have been introduced specifically for in strategic reasoning such as Strategy Logic (SL) (Chatterjee, Henzinger, and Piterman 2010; Mogavero et al. 2014). This has lead to investigating the "*behavioral*" semantics in these logics. In their seminal work (Mogavero et al. 2014), Mogavero et al. introduce and analyze the behavioral aspects of quantification in SL: a logic for reasoning about the strategic behavior of agents in a context where the properties of executions are expressed in LTL. They show that restricting to behavioral quantification of strategies is a way of both making the semantics more realistic and computationally easier. In addition, they proved that behavioral and non-behavioral semantics coincide for certain fragments, including the one corresponding to the well known ATL* (Alur, Henzinger, and Kupferman 2002), but diverge for more interesting classes of formulas, e.g., the ones that can express game-theoretic properties such as Nash Equilibria and the like. This has started a new line of research that aims at identifying new notions of behavioral and non-behavioral quantification, as well as characterize the syntactic fragments that are invariant to these semantic variations (Gardy, Bouyer, and Markey 2018; Gardy, Bouyer, and Markey 2020).

In this paper we introduce a behavioral semantics for QLTL. The resulting logic, called *Behavioral*-QLTL (QLTL_B) is characterized by the fact that the functions that assign the truth value of the quantified propositions along the trace can only depend on the past. In other words such functions must be "*processes*". This makes QLTL_B perfectly suitable to capture extended forms of planning through standard reasoning tasks (satisfiability in particular).

Indeed, temporally extended planning in nondeterministic domains, as well as LTL synthesis, are expressed in QLTL_B through formulas with a simple quantification alternation. While, as this alternation increases, we get to forms of planning/synthesis in which conditional and conformant planning aspects get mixed. For example, the QLTL_B formula of the form $\exists Y \forall X \psi$ represents the conformant planning over the LTL specification (of both environment model and goal) ψ, as it is intended in (Rintanen 2004) (note that this could be done also with standard QLTL, since $\exists Y$ is put upfront as it cannot depend on the nondeterministic evolution of the fluents in the

[*] Joint work with Giuseppe De Giacomo.

planning domain). Instead, the QLTL$_\mathsf{B}$ formula $\forall X \exists Y \psi$ represents contingent planning, i.e., *(Strong) Planning in Fully Observable Nondeterministic Domains* (FOND), as well as LTL synthesis (which, instead, could not be captured in standard QLTL). By taking QLTL$_\mathsf{B}$ formulas with increased alternation, one can describe more complex forms of planning and synthesis. The QLTL$_\mathsf{B}$ formula $\forall X_1 \exists Y \forall X_2 \varphi$ represents the problem of *(Strong) Planning in Partially Observable Nondeterministic Domains* (POND), where X_1 and X_2 are the visible and hidden parts of the domain, respectively. By going even further in alternation, we get a generalized form of POND where a number of actuators with hierarchically reduced visibility are coordinated to execute a plan that fulfills a temporally extended goal in an environment model. Interestingly this instantiates problems of distributed synthesis with hierarchical information studied in formal methods (Pnueli and Rosner 1990; Kupferman and Vardi 2001; Finkbeiner and Schewe 2005).

We study QLTL$_\mathsf{B}$, by introducing a formal semantics that is *Skolem-based*, meaning that we assign existential values through Skolem-like functions that depend on the universal (adversarial) choice of the variables of interest. Specifically we restrict sich Skolem function to depend only on the past and hence behave as processes/strategies/plans. As a matter of fact, such Skolem functions can be represented as suitable labeled trees, describing all the possible executions of a given process that receive inputs from the environment. We characterize the complexity of satisfiability in QLTL$_\mathsf{B}$ as $(n+1)$-EXPTIME-complete, with n being the number of quantification blocks of the form $\forall X_i \exists Y_i$ the in the formula. This improves the complexity of satisfiability for classic QLTL, which depends on the overall quantifier alternation in the formula, and in particular is $2(n-1)$-EXSPACE-complete. Moreover, it also shows that the corresponding synthesis and planning problems can be optimally solved in QLTL$_\mathsf{B}$, as the matching lower-bound is provided by a reduction of these problems.

We also consider a weak variant of QLTL$_\mathsf{B}$, called *Weak Behavioral*-QLTL (QLTL$_\mathsf{WB}$), where the history is always visible while we have restriction visibility on the curent instant only. We show that the complexity of satisfiability in QLTL$_\mathsf{WB}$ is 2-EXPTIME-complete, regardless of the number and alternation of quantifiers. The reason for this is in that processes are modeled in a way that they have full visibility on the past computation. This allows them to find the right plan by means of a local reasoning, and so without employing computationally expensive automata projections. As for the case of QLTL$_\mathsf{B}$, such procedure is optimal to solve the corresponding synthesis problems, as the matching lower-bound is again provided by a reduction of them.

Acknowledgements

Work supported by the ERC Advanced Grant WhiteMech (No. 834228) and by the EU ICT-48 2020 project TAILOR (No. 952215).

References

Abadi, M.; Lamport, L.; and Wolper, P. 1989. Realizable and unrealizable specifications of reactive systems. In *ICALP'89*, volume 372 of *LNCS*, 1–17. Springer.

Alur, R.; Henzinger, T.; and Kupferman, O. 2002. Alternating-Time Temporal Logic. *JACM* 49(5):672–713.

Aminof, B.; De Giacomo, G.; Murano, A.; and Rubin, S. 2019. Planning under LTL environment specifications. In *ICAPS*, 31–39. AAAI Press.

Bacchus, F., and Kabanza, F. 1998. Planning for Temporally Extended Goals. *Ann. Math. Artif. Intell.* 22(1-2):5–27.

Bacchus, F., and Kabanza, F. 2000. Using Temporal Logics to Express Search Control Knowledge for Planning. *Artif. Intell.* 116(1-2):123–191.

Barringer, H.; Fisher, M.; Gabbay, D. M.; Gough, G.; and Owens, R. 1995. METATEM: an introduction. *Formal Aspects Comput.* 7(5):533–549.

Bertoli, P.; Cimatti, A.; and Roveri, M. 2001. Heuristic search + symbolic model checking = efficient conformant planning. In *IJCAI'01*, 467–472.

Calvanese, D.; De Giacomo, G.; and Vardi, M. Y. 2002. Reasoning about Actions and Planning in LTL Action Theories. In *KR'02*, 593–602.

Camacho, A.; Bienvenu, M.; and McIlraith, S. A. 2019. Towards a Unified View of AI Planning and Reactive Synthesis. In *ICAPS'19*, 58–67.

Cerrito, S., and Mayer, M. 1998. Bounded Model Search in Linear Temporal Logic and Its Application to Planning. In *TABLEAUX'98*, volume 1397 of *LNCS*, 124–140. Springer.

Chatterjee, K., and Henzinger, T. A. 2007. Assume-Guarantee Synthesis. In *TACAS'07*, volume 4424 of *LNCS*, 261–275.

Chatterjee, K.; Henzinger, T. A.; and Jobstmann, B. 2008. Environment Assumptions for Synthesis. In *CONCUR'08*, 147–161.

Chatterjee, K.; Henzinger, T. A.; and Piterman, N. 2010. Strategy logic. *Inf. Comput.* 208(6):677–693.

Church, A. 1963. Logic, arithmetics, and automata. In *Proc. Int. Congress of Mathematicians, 1962*, 23–35.

Cimatti, A., and Roveri, M. 2000. Conformant Planning via Symbolic Model Checking. *J. Artif. Intell. Res.* 13:305–338.

Cimatti, A.; Giunchiglia, F.; Giunchiglia, E.; and Traverso, P. 1997. Planning via model checking: A decision procedure for *AR*. In *ECP'97*, volume 1348 of *LNCS*, 130–142.

Daniele, M.; Traverso, P.; and Vardi, M. Y. 1999. Strong cyclic planning revisited. In *ECP'99*, volume 1809 of *LNCS*, 35–48. Springer.

Finkbeiner, B., and Schewe, S. 2005. Uniform distributed synthesis. In *LICS'05*, 321–330.

Gardy, P.; Bouyer, P.; and Markey, N. 2018. Dependences in Strategy Logic. In *STACS'18*, volume 96 of *LIPIcs*, 34:1–34:15.

Gardy, P.; Bouyer, P.; and Markey, N. 2020. Dependences in strategy logic. *Theory Comput. Syst.* 64(3):467–507.

Green, C. C. 1969. Application of theorem proving to problem solving. In *IJCAI'69*, 219–240.

Kupferman, O., and Vardi, M. Y. 2001. Synthesizing distributed systems. In *LICS'01*, 389–398.

Mogavero, F.; Murano, A.; Perelli, G.; and Vardi, M. Y. 2014. Reasoning about strategies: On the model-checking problem. *ACM TOCL* 15(4):34:1–34:47.

Pnueli, A., and Rosner, R. 1989. On the Synthesis of a Reactive Module. In *POPL*, 179–190. ACM.

Pnueli, A., and Rosner, R. 1990. Distributed reactive systems are hard to synthesize. In *FOCS'90*, 746–757.

Rintanen, J. 2004. Complexity of Planning with Partial Observability. In *ICAPS'04*, 345–354.

Sistla, A. P. 1985. *Theoretical Issues in the Design and Verification of Distributed Systems.* Ph.D. Dissertation.