Categorical Semantics for Counterpart-based Temporal Logics in Agda*

Fabio Gadducci, Andrea Laretto, and Davide Trotta

Department of Computer Science, University of Pisa, Italy
fabio.gadducci@unipi.it, andrea.laretto@gmail.com, trottadavide92@gmail.com

Abstract

In this work we present the categorical semantics of first-order temporal logics, providing our models by means of relational presheaves and adopting the perspective of counterpart semantics. We also illustrate a computer-assisted formalization of these constructions using the proof assistant Agda, highlighting the crucial aspects of our formalization and the practical use of (quantified) temporal logics in a constructive proof assistant. We employ the `agda-categories` library to capture the notions of category theory in our setting.

Introduction

Temporal logics have proven to be one of the most well-established and effective techniques for the verification of both large-scale and stand-alone programs. Research on temporal logics, among many other aspects, focused on sufficiently expressive fragments of logics that are suitable for the specification of complex multi-component systems. A prominent use case is for example that of graph logics [1, 2], where states are specialized as graphs and transitions are families of (partial) graph morphisms. These logics combine temporal and spatial reasoning and allow, for example, to express the possible transformation of a graph topology over time.

One of the defining characteristics of graph logics is that they permit reasoning and expressing properties on the individual elements of the graph, thus suggesting for the use of (first-order) quantified temporal logics. In order to account for the evolution of elements in time, in many practical applications the models under consideration need to be able to adequately express the creation, dellocation, merging, and possibly duplication of elements. Take, for example, the scenario of processes being allocated and freed in memory. A possible approach to the representation of these models is the *counterpart paradigm* proposed by Lewis [3]: each state of the transition system identifies a local set of elements, and possibly partial morphisms carry the identity of elements from one state to the other. This perspective is applied in [4] to model a counterpart-based μ-calculus with second order quantifiers, and it is generalized to a categorical setting in [5] using *relational presheaves* by building on the ideas of [6].

In this work we describe a categorical semantics for a first-order linear temporal logic QLTL that can reason about the temporal evolution of many-sorted algebras, along with a formalization of this logic and its semantics in Agda. For a complete overview of these results, we refer to [7]. A formal presentation of modal and temporal logics in a proof assistant effectively provides a playground in which the mechanisms and the validity of these logics can be expressed, tested, and experimented with. Moreover, given the constructive interpretation of the Agda code, our formalization essentially codifies a procedure to convert classical set-theoretical notions into categorical ones in the setting of temporal logics, further showing the correctness and coherence of the ideas presented by previous authors in [5]. This work also establishes the usefulness and flexibility of the `agda-categories` library from the point of view of practical applications. The Agda formalization is available at `https://github.com/iwilare/categorical-qtl`.

*Research partially supported by the Italian MIUR project PRIN 2017FTXR7S "IT-MaTTerS".

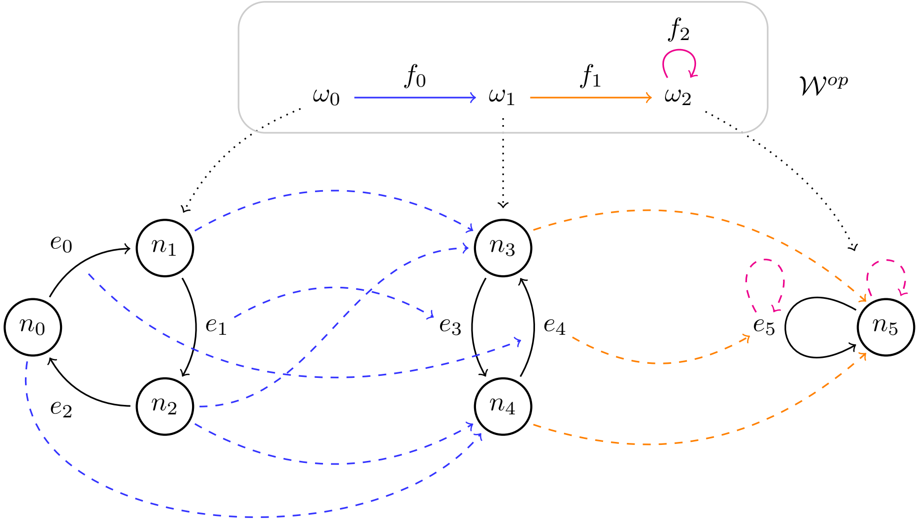

Figure 1: An algebraic counterpart $\mathcal{W}$-model and its corresponding category $\mathcal{W}^{op}$.

Counterpart Semantics via Relational Presheaves

We now introduce the models of our logic by sketching the notion of *algebraic counterpart $\mathcal{W}$-model* on a given many-sorted algebraic signature Σ. Our models provide the following: a category $\mathcal{W}$, where each object represents a world in a given instant of time, and morphisms represent temporal evolutions; for each sort $\tau \in \text{Sorts}(\Sigma)$, a so-called *relational presheaf* $[\![\tau]\!] : \mathcal{W}^{op} \to \mathbf{Rel}$ is provided, where **Rel** is the category of sets and relations. Such a presheaf assigns to each world a set of individuals and to each morphism a corresponding relation between worlds. Following the counterpart paradigm, the individuals of two worlds are related by a relation R when they are the same individual after a temporal evolution, and we say that the individual in the next world *is a counterpart of* the individual in the previous one. Finally, opportune morphisms of relational presheaves are given to represent the set functions of the algebras in each world. A graphical representation of a model is shown in Figure 1 by taking as example the signature of graphs **Gr**, where we indicate the graph edges with solid arrows and the counterpart relations with dashed arrows, without distinguishing the two relations connected the nodes and edges of the algebras.

We now give the syntax of our quantified (linear) temporal logic QLTL on the previously defined models. Given a set of denumerable variables $\mathcal{X}$ with $x \in \mathcal{X}$, the set $\mathcal{A}^{\mathsf{QLTL}}$ of QLTL formulae-in-context is given by

$$\phi := \mathsf{true} \mid m =_\tau n \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid \exists_\tau x.\phi \mid \mathsf{O}\phi \mid \phi_1 \mathsf{U} \phi_2 \mid \phi_1 \mathsf{W} \phi_2,$$

where m and n are terms on the Σ-algebra with the same sort $\tau \in \text{Sorts}(\Sigma)$. The *next* operator $\mathsf{O}\phi$ expresses the fact that a given individual has at least one counterpart at the next state where ϕ holds, and the *until* operator $\phi_1 \mathsf{U} \phi_2$ indicates that the counterparts of an individual exist and satisfy the property ϕ_1 until a point in time where a counterpart satisfies ϕ_2.

In our example, we can express properties on both the structure and the temporal evolution of the graphs. For instance, we define the following formulae about existence of nodes and loops in a graph:

$$\begin{array}{rcl}
\mathbf{loop}(e) & := & s(e) =_{\mathrm{Node}} t(e), \\
\mathbf{present}_{\tau}(x) & := & \exists_{\tau} y . x =_{\tau} y, \\
\mathbf{nextStepDeallocated}_{\tau}(x) & := & \mathbf{present}_{\tau}(x) \wedge \neg \mathsf{O} \mathbf{present}_{\tau}(x), \\
\mathbf{nodeHasLoop}(n) & := & \exists_{\mathrm{Edge}} e . s(e) =_{\mathrm{Node}} n \wedge \mathbf{loop}(e), \\
\mathbf{willBecomeLoop}(e) & := & \neg \mathbf{loop}(e) \wedge \mathbf{present}_{\mathrm{Edge}}(e) \, \mathsf{U} \, \mathbf{loop}(e),
\end{array}$$

and we have that the following sets of individuals satisfy the given formulae in some worlds:

$$\begin{array}{ll}
\llbracket [e : \mathrm{Edge}] \; \mathbf{nextStepDeallocated}(e) \rrbracket_{w_0} & = \{e_2\}, \\
\llbracket [n : \mathrm{Node}] \; \mathbf{nodeHasLoop}(n) \rrbracket_{w_2} & = \{n_5\}, \\
\llbracket [e : \mathrm{Edge}] \; \mathbf{willBecomeLoop}(e) \rrbracket_{w_0} & = \{e_0\}.
\end{array}$$

Positive normal forms

Our Agda implementation enables the possibility to prove metatheorems and construct algorithms that can operate on the semantics of the logic and its models. One example of such a metatheorem is the investigation of a suitable *positive normal form* of this logic, which we also formalize in Agda by considering both the case of counterpart relations as partial functions and as general (possibly duplicating) relations. This additional development is available at `https://github.com/iwilare/qltl-pnf`. Aside from the standard applications in the setting of temporal model checking, such a presentation is crucial in our setting of *constructive* proof assistants due to the intuitionistic interpretation of negation: first, because the lack of classical reasoning limits the expressiveness of our logic to just its *intuitionistic fragment*, and for example we cannot prove in Agda that the QLTL formula $\neg\neg\phi \implies \phi$ is validated for every possible choice of model and subformula ϕ; second, because negation in subformulae can be particularly tricky to deal with, as it forces the user to prove the validity of formulae with a *reductio-ad-absurdum* approach instead of having to show that a direct formula holds.

Conclusion and Future work

In our talk we will introduce the most salient aspects of the formalization of our logic QLTL and its use in a proof assistant, along with a brief presentation of its categorical semantics.

Embedding our temporal logic in a proof assistant also enables the user to leverage the *assistant* part of the tool. This can be done by the use of either *internal* or *external* solvers: in the first case, one defines (verified) model checking algorithms in Agda itself by restricting to models where each counterpart relation is finite, possibly by also exploiting reflection mechanisms [8]; in the second case, the implementer of the logic writes bindings to connect the proof assistant with external programs, such as model checkers or SMT and SAT solvers, so that proving the formula or providing a counterexample is offloaded to a more efficient and specialized program [9]. A possible extension of this work could be the implementation of either of these mechanisms to the setting of counterpart models and their semantics.

References

[1] B. Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990.

[2] A. Dawar, P. Gardner, and G. Ghelli. Expressiveness and complexity of graph logic. *Information and Computation*, 205(3):263–310, 2007.

[3] D. K. Lewis. Counterpart theory and quantified modal logic. *The Journal of Philosophy*, 65(5):113–126, 1968.

[4] F. Gadducci, A. Lluch-Lafuente, and A. Vandin. Counterpart semantics for a second-order μ-calculus. *Fundamenta Informaticae*, 118(1-2):177–205, 2012.

[5] F. Gadducci and D. Trotta. A presheaf semantics for quantified temporal logics. *CoRR*, abs/2111.03855, 2021. arXiv: 2111.03855.

[6] S. Ghilardi and G. Meloni. Modal and tense predicate logic: Models in presheaves and categorical conceptualization. In F. Borceux, editor, *Categorical Algebra and its Applications*, volume 1348 of *Lecture Notes in Mathematics*, pages 130–142. Springer, 1988.

[7] A. Laretto. Counterpart semantics for quantified temporal logics: sets, categories and Agda. Master's thesis, University of Pisa, October 2022. `https://iwilare.com/msc-thesis.pdf`.

[8] J. Esparza, P. Lammich, R. Neumann, T. Nipkow, A. Schimpf, and J. Smaus. A fully verified executable LTL model checker. In *Computer Aided Verification*, pages 463–478. Springer Berlin Heidelberg, 2013.

[9] G. Shen and L. Kuper. Toward SMT-based refinement types in Agda. *CoRR*, abs/2110.05771, 2021. arXiv: 2110.05771.