Positive normal forms for counterpart-based temporal logics

Fabio Gadducci
Department of Computer Science, University of Pisa, Pisa, IT

Andrea Laretto
Department of Computer Science, University of Pisa, Pisa, IT

Davide Trotta
Department of Computer Science, University of Pisa, Pisa, IT

Abstract

Temporal logics provide an expressive formalism for the verification of computer systems and their evolution in time. Quantified variations of these logics allow to reason in terms of the individual constituents of the system, thereby increasing their expressiveness and generality. A key property for these logics is the presence of positive normal form of formulae, since it allows to provide a fixpoint-based semantics and to simplify automated model checking.
The paper considers quantified linear time logics and, in order to deal with the trans-world identity problem, it introduces a counterpart semantics to relate the elements of the system as it evolves in time. A positive normal form of formulae is then discussed: in the case of counterpart semantics, the positive normal form turns out to be a non-trivial transformation that noticeably alters the logics in question, which we investigate in detail in this work.

2012 ACM Subject Classification Theory of computation → Higher order logic

Keywords and phrases Counterpart semantics, quantified temporal logics, positive normal form

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

1 Introduction

The use of temporal logics for the specification and verification of computational systems is well-established and has proved extremely effective both in the context of stand-alone programs and large-scale systems, see for example [18, 4] among many others. After the foundational work by Pnueli [17], the research on these topics has focused on both algorithmic procedures for the verification of properties as well as finding sufficiently expressive fragments of temporal logics suitable for the specification of complex multi-component systems. Over the years, several models for temporal logics have been developed. Transition systems (also known as Kripke frames) can be considered the leading example of such models for temporal logics. In a transition system, each state represents a configuration of the device and each transition identifies a possible state evolution. Depending on the setting one is considering, states and transitions are usually specialized and enriched with algebraic structures. A prominent example regards models of graph logics [5, 6, 7], where states are graphs and transitions are families of (partial) graph morphisms. To this end, the concept of quantified temporal logics has been proposed as a sufficiently expressive formalism that allows to refer to the single elements of the system, despite the undecidability of these logics [9, 13].

From a theoretical point of view, the semantical models for such logics are not clearly cut. Consider for example a simple model with two states s_0, s_1, two transitions $s_0 \to s_1$ and $s_1 \to s_0$, and an item i that appears in s_0. Is the item i being destroyed and recreated again and again, or is it just an identifier that is being reused? The issue is denoted in the literature as the *trans-world identity problem*, [12, 1] and the typical solution provided by the so-called "Kripke semantics" is to fix a set of universal items that gives identity to each element in the state. However, Kripke-like solutions are not fully adequate to model systems with dynamic allocation and deallocation of components: in the example above, every i is the same after each deallocation since it belongs to the universal domain. But intuitively, every instance of i should be considered as distinct, despite being syntactically equivalent.

The proposal advanced by Lewis [15] is the *counterpart paradigm*: instead of a universal set of items, each state identifies a local set of elements, and (partial) morphisms connect them by carrying elements from one state to the other, allowing for deallocation and merging of items. The flexibility of these models allows to properly deal with the trans-world identity problem by letting us speak formally about entities that are destroyed and (re)created.

Besides its clear foundational appeal, the relevance of the counterpart solution has to be validated by the possible application to the search of feasible algorithms for satisfiability. To this end, positive normal forms (i.e., where negation is defined only for atomic propositions) are a standard tool of temporal logics, since they can be used to simplify both the theoretical treatment of such logics as well as facilitating model checking algorithms from a practical point of view [14, 3]. Additionally, the positive normal form is especially useful when presenting a fixpoint-based semantics of the logic and ensure its well-definition, as it excludes the notion of negation and allows monotonicity to be easily shown.

The main purpose of this work is to provide a counterpart-style semantics and a presentation in positive normal form for a quantified linear temporal logic, denoted QLTL, and for a strictly more expressive extension $\widehat{\mathsf{QLTL}}$. In particular, after introducing in detail their syntax and semantics, we define two new logics, presented in positive normal form style and denoted by PNF and $\widehat{\mathsf{PNF}}$ respectively, and their semantics. Then we prove a semantical equivalence between QLTL and PNF, and similarly for $\widehat{\mathsf{QLTL}}$ and $\widehat{\mathsf{PNF}}$. This allows us to formally conclude that, with respect to the counterpart semantics, PNF is precisely the positive normal form presentation of QLTL, and that $\widehat{\mathsf{PNF}}$ is precisely the positive normal form presentation of $\widehat{\mathsf{QLTL}}$.

The counterpart-based temporal logics QLTL and $\widehat{\mathsf{QLTL}}$ we consider are inspired by the previous work of [10, 11]. The presentation given here has been greatly simplified in order to focus on the mechanisms required to obtain the positive normal form, while still aiming for sufficient expressivity. In particular, we restrict ourselves to a first-order fragment in which the states are considered not as algebras, but as standard Kripke-style sets of elements; however, the minimal fragment of the logic presented here can easily be re-extended to a richer presentation while still maintaining the results on the positive normal form.

Section 2 introduces our counterpart models, while Section 3 presents the syntax and the semantics of our quantified temporal logics. Section 4 discusses positive normal forms, and proves various equivalences among different operators. Finally, Section 5 wraps up the paper and discusses possible future works. We formalised the proofs of the main results presented in this work using the dependently-typed language and interactive proof assistant Agda [16]. The mechanization can be retrieved at `https://github.com/iwilare/qltl-pnf`.

2 Counterpart semantics

In this section we define the notions and class of models on which our quantified logic is interpreted. We start by recalling the notion of *Kripke frame* as widely known in modal logics [1, 2] and extend it for the case of counterpart semantics.

▶ **Definition 1.** *A* ***Kripke frame*** *is a 3-tuple* $\langle W, R, D\rangle$ *defined as*

- W *is a non-empty set;*
- R *is a binary relation on* W*;*

- *D is a function assigning to every element $\omega \in W$ a non-empty set $D(w)$ such that if $\omega R \omega'$ then $D(\omega) \subseteq D(\omega')$.*

The set W is intuitively interpreted as the domain of all *possible worlds*, whereas the binary relation R represents an *accessibility relation* among worlds, connecting them whenever a transition from a world to another is possible. The *domain* $D(\omega)$ identifies the set of objects that are present in the world ω.

A crucial development in the presentation of Kripke models was introduced by Lewis [15] with the notion of *counterpart relations* and the subsequent introduction of counterpart theory. The idea is to tackle the trans-world identity problem by rejecting strict identity of individuals, and instead employing the notion of *counterpart relation* between worlds.

Inspired by Lewis's approach, a more general notion of counterpart model is considered in [10], where worlds are related through *multiple* accessibility relations, and each accessibility relation is equipped with a proper counterpart relation.

▶ **Definition 2.** *A* ***counterpart model*** *is a 3-tuple $\langle W, D, \mathcal{C}\rangle$ defined as*
- *W and D are defined as for Kripke frames;*
- *$\mathcal{C}$ is a function assigning to every 2-tuple $\langle \omega, \omega' \rangle$ a set $\mathcal{C}\langle \omega, \omega' \rangle \in \wp(D(\omega) \rightharpoonup D(\omega'))$, where $\wp$ denotes the powerset, and every element $C \in \mathcal{C}\langle \omega, \omega' \rangle$ is a partial function. We call these partial functions* ***atomic (or one-step) counterpart functions****.*

Given two worlds ω and ω', the set $\mathcal{C}\langle \omega, \omega' \rangle$ is the collection of atomic transitions from ω to ω', defining the possible ways we can access worlds with a *one-step transition* in the system. When the set $\mathcal{C}\langle \omega, \omega' \rangle$ is empty, there are no atomic transitions from ω to ω'.

Each atomic counterpart function $C \in \mathcal{C}\langle \omega, \omega' \rangle$ connects the individuals between two given worlds ω and ω', intuitively identifying them as the same element after one time evolution of the model. In particular, if we consider two elements $s \in D(\omega)$ and $s' \in D(\omega')$ and a function $C \in \mathcal{C}\langle \omega, \omega' \rangle$, if $C(s)$ is defined and $C(s) = s'$ then s' represents a future development of s via C.

The use of partial functions in the counterpart model allows us to model the notion of removal of an element from the system, so that there exists no counterpart in the next state. For example, if there is no element $s' \in D(\omega')$ such that $s' = C(s)$, then we can conclude that the element s has been deallocated by C.

Now we formally introduce counterpart functions, fixing notation for the rest of the paper.

▶ **Definition 3.** *A partial function C of $D(\omega) \rightharpoonup D(\omega')$ is a* ***counterpart function*** *if one of the following three cases holds: C is the identity function, $C \in \mathcal{C}\langle \omega, \omega' \rangle$ is a one-step counterpart function given by the model, or C can be obtained by composing together a suitable sequence of functions $C_n \circ \cdots \circ C_0$ with $C_i \in \mathcal{C}\langle \omega_i, \omega_{i+1} \rangle$.*

We remark here that the resulting composition $C_2 \circ C_1 : D(\omega_1) \rightharpoonup D(\omega_3)$ of two atomic counterpart functions $C_1 \in \mathcal{C}\langle \omega_1, \omega_2 \rangle$ and $C_2 \in \mathcal{C}\langle \omega_2, \omega_3 \rangle$ might not necessarily be an atomic counterpart function. This intuitively represents the fact that transitioning through an intermediate state and transitioning directly between worlds can be regarded as two different possibilities, with the model defining only the direct one-step transitions.

▶ **Definition 4.** *We say that an individual $s' \in D(\omega')$* ***is the counterpart of*** *$s \in D(\omega)$ through a counterpart function C if $C(s)$ is defined and $C(s) = s'$.*

▶ **Example 5** (Counterpart model). In Figure 1 we provide a graphical presentation of the counterpart model defined by the set of worlds $W := \{\omega_1, \omega_2, \omega_3\}$, where for example $D(\omega_0) = \{a_0, b_0, c_0\}$, $D(\omega_1) = \{a_1, b_1, c_1, d_1\}$, and $D(\omega_2) = \{a_2, b_2, c_2, d_2\}$. The worlds are connected by the following functions: $\mathcal{C}\langle\omega_0,\omega_1\rangle := \{C_0\}$ a single counterpart function C_0 between ω_0 and ω_1, $\mathcal{C}\langle\omega_1,\omega_2\rangle := \{C_1, C_2\}$ has two possible counterpart functions between ω_1, ω_2, and $\mathcal{C}\langle\omega_2,\omega_2\rangle = \{C_3\}$ is a looping counterpart function. Note that we use solid and dashed lines to distinguish C_1 and C_2, respectively.

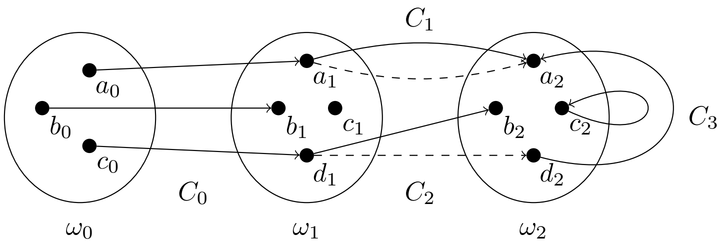

Figure 1 An example of counterpart model.

As is the case of LTL where we can identify traces connecting linearly evolving states, see for example [2], we can consider linear sequences of counterpart functions providing a list of sequentially accessible worlds.

▶ **Definition 6.** *A **trace** σ on a counterpart model $\langle W, D, \mathcal{C}\rangle$ is an infinite sequence of one-step counterpart functions $(C_0, C_1, \dots)$ such that $C_i \in \mathcal{C}\langle\omega_i, \omega_{i+1}\rangle$ for any $i \geq 0$.*

Given a trace $\sigma = (C_0, C_1, \dots)$, we use i as subscript $\sigma_i := (C_i, C_{i+1}, \dots)$ to denote the trace obtained by excluding the first i counterpart functions. We use $\sigma_\bullet$ to indicate the first world ω_0 of the trace σ.

Since a trace $\sigma = (C_0, C_1, \dots)$ provides a sequence of counterpart functions step-by-step connected through a world, we denote with $C_{\leq i}$ the composite function $C_{i-1} \circ \dots \circ C_0$ from the first world ω_0 up to the i-th world ω_i through the functions given by the trace σ. In the special case $i = 0$, the function $C_{\leq 0}$ is defined to be the identity function on ω_0.

3 Quantified linear temporal logics

In this section we present the syntax and semantics of our (first-order) quantified linear temporal logic QLTL and its extension $\widehat{\mathsf{QLTL}}$. We will assume hereafter a fixed counterpart model $\langle W, D, \mathcal{C}\rangle$, with definitions referring to the data provided by the underlying model.

3.1 Syntax and semantics of QLTL

In order to provide a simpler presentation, it is customary to exclude all constructs that can be expressed in terms of other operators, such as conjunction and universal quantification; for this reason, we will initially present QLTL with a minimal set of operators and derive other ones with negation.

▶ **Definition 7** (QLTL)**.** *Let $\mathcal{X}$ be a set of variables with $x, y \in \mathcal{X}$, and P a set of unary predicates. The set $\mathcal{F}^{\mathsf{QLTL}}$ of QLTL formulae is generated by the following rules:*

$$\psi := \mathsf{tt} \mid x = y \mid P(x), \qquad \phi := \psi \mid \neg\phi \mid \phi \vee \phi \mid \exists x.\phi \mid \mathsf{O}\psi \mid \phi_1 \mathsf{U} \phi_2.$$

The *next* operator $\mathsf{O}\phi$ expresses the fact that a certain property ϕ has to be true at the next state. The *until* operator $\phi_1 \mathsf{U} \phi_2$ indicates that the property ϕ_1 has to hold at least until the property ϕ_2 becomes true, which must hold at the present or future time.

We use the letter ψ to indicate the case of elementary predicates and we refer to these formulae as *atomic formulae*. Given two variables $x, y \in \mathcal{X}$ denoting two individuals, the formula $x = y$ indicates that the two individuals coincide in the current world. Our logic is extended with a unary predicate symbol $P(x)$ that will be used in the running example in Figure 2. The usual dual operators can be syntactically expressed by taking $\mathsf{ff} := \neg\mathsf{tt}$, $\phi_1 \wedge \phi_2 := \neg(\neg\phi_1 \vee \neg\phi_2)$, and $\forall x.\phi := \neg\exists x.\neg\phi$.

As we anticipated in Section 2, one of the main advantages of a counterpart semantics is the possibility to reason about deallocating and merging elements of a system. For example, we can capture a notion of existence of an element at the current moment with the shorthand $\mathbf{present}(x) := \exists y.x = y$. We can then combine this predicate with the *next* operator to talk about elements that are present in the current world and that will still be present at the next step, for example with the formula $\mathbf{nextStepPreserved}(x) := \mathbf{present}(x) \wedge \mathsf{O}\mathbf{present}(x)$. Similarly, we can refer to elements that are now present but that will be deallocated at the next step by considering $\mathbf{nextStepDeallocated}(x) := \mathbf{present}(x) \wedge \neg\mathsf{O}\mathbf{present}(x)$.

Since free variables referring to individuals can now appear inside formulae, we recall the usual presentation of context and formulae-in-context as similarly defined in [11, 10].

▶ **Definition 8** (Context). *A context Γ over a set of variables $\mathcal{X}$ is a subset of $\mathcal{X}$. We use the notation Γ, x to indicate the augmented context $\Gamma \cup \{x\}$.*

▶ **Definition 9** (Formulae-in-context). *A formula-in-context is a formula ϕ along with an associated context Γ in which it is defined, and we indicate this decoration with $[\Gamma]\phi$. Any context is such that $\mathrm{fv}(\phi) \subseteq \Gamma$, i.e., it contains the free variables of the formula ϕ.*

We omit the bracketed context whenever it is unnecessary to specify it.

To properly present the notion of satisfiability of a formula with respect to a given counterpart model, we need to first introduce the definition of *assignment* in a given world.

▶ **Definition 10** (Assignment). *An* ***assignment*** *in the world $\omega \in W$ for the context Γ is a function $\mu : \Gamma \to D(\omega)$. We use the notation $\mathcal{A}_\omega^\Gamma$ to indicate the set of assignments μ defined in ω for the context Γ.*

Moreover, we denote by $\mu[x \mapsto s]$ the assignment with type $\Gamma, x \to D(\omega)$ obtained by extending the domain of μ with $s \in D(\omega)$ at the variable $x \notin \Gamma$.

We now define the lifting of counterpart functions to assignments. The intuition behind this notion is that we want to translate all elements of an assignment to the next world using the counterpart function individual-by-individual.

▶ **Definition 11** (Counterpart functions on assignments). *Given a counterpart function $C : D(\omega_1) \rightharpoonup D(\omega_2)$ and an assignment $\mu : \Gamma \to D(\omega_1)$, we say that the assignment $C \circ \mu : \Gamma \to D(\omega_2)$* ***is defined*** *whenever $C(\mu(x))$ is defined for all variables $x \in \Gamma$.*

We now introduce the notion of satisfiability of a formula in a given trace and assignment.

▶ **Definition 12** (QLTL satisfiability). *Given a QLTL formula-in-context $[\Gamma]\phi$, a trace $\sigma = (C_0, C_1, \dots)$, and an assignment $\mu : \Gamma \to D(\sigma_\bullet)$ defined in the first world of σ, we present the satisfiability relation as follows:*

- $\sigma, \mu \vDash \mathsf{tt}$;
- $\sigma, \mu \vDash x = y$ *if* $\mu(x) = \mu(y)$;
- $\sigma, \mu \vDash P(x)$ *if* $P(\mu(x))$;
- $\sigma, \mu \vDash \neg\phi$ *if* $\sigma, \mu \nvDash \phi$;
- $\sigma, \mu \vDash \phi_1 \vee \phi_2$ *if* $\sigma, \mu \vDash \phi_1$ *or* $\sigma, \mu \vDash \phi_2$;

- *$\sigma, \mu \vDash \exists x.\phi$ if there exists an individual $s \in D(\sigma_\bullet)$ such that $\sigma, \mu[x \mapsto s] \vDash \phi$;*
- *$\sigma, \mu \vDash \mathsf{O}\phi$ if $C_0 \circ \mu$ is defined and $\sigma_1, C_0 \circ \mu \vDash \phi$;*
- *$\sigma, \mu \vDash \phi_1 \mathsf{U} \phi_2$ if there exists an $\bar{n} \geq 0$ such that:*
  1. *for any $i < \bar{n}$, we have that $C_{\leq i} \circ \mu$ is defined and $\sigma_i, C_{\leq i} \circ \mu \vDash \phi_1$;*
  2. *$C_{\leq \bar{n}} \circ \mu$ is defined and $\sigma_{\bar{n}}, C_{\leq \bar{n}} \circ \mu \vDash \phi_2$.*

▶ **Example 13.** We present a running example in Figure 2 to better describe the expressiveness of QLTL and to illustrate the mechanisms of working in a counterpart-based semantics.

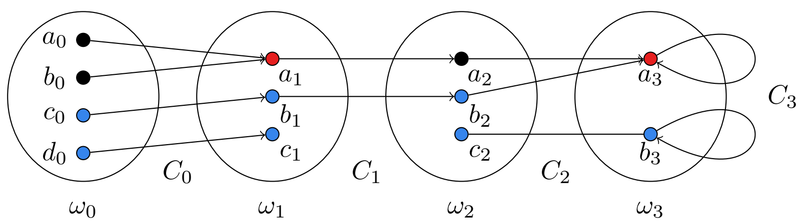

Figure 2 An example with four worlds $\omega_0, \omega_1, \omega_2, \omega_3$

We consider a fixed trace $\sigma = (C_0, C_1, C_2, C_3, C_3, \dots)$ and we indicate with $\mathsf{B}(x)$ and $\mathsf{R}(x)$ the unary predicates that hold for any individual coloured in blue and red, respectively. As a concrete scenario for the temporal operators $\mathsf{O}\phi$ and $\phi_1 \mathsf{U} \phi_2$ we presented in Definition 12, we have for example that $\sigma, \{x \mapsto a_0\} \vDash \mathsf{O}(\mathsf{R}(x))$, and $\sigma, \{x \mapsto c_0\} \vDash \mathsf{B}(x)\mathsf{U}\mathsf{R}(x)$. Also, we have that a_0 is preserved at the next step with $\sigma, \{x \mapsto a_0\} \vDash \mathbf{nextStepPreserved}(x)$, whereas c_1 is removed and indeed $\sigma_1, \{x \mapsto c_1\} \vDash \mathbf{nextStepDeallocated}(x)$.

▶ Remark 14 (*Eventually* operator $\Diamond\phi$). As in LTL, we can define an additional *eventually* $\Diamond\phi$ operator as $\Diamond\phi := \mathsf{tt}\mathsf{U}\phi$. Its semantics can be presented directly as

- *$\sigma, \mu \vDash \Diamond\phi$ if there exists $i \geq 0$ such that $C_{\leq i} \circ \mu$ is defined and $\sigma_i, C_{\leq i} \circ \mu \vDash \phi$.*

In our example in Figure 2, we have for instance that $\sigma, \{x \mapsto c_0\} \vDash \Diamond\mathsf{R}(x)$ but $\sigma, \{x \mapsto d_0\} \nvDash \Diamond\mathsf{R}(x)$ and similarly $\sigma_2, \{x \mapsto c_2\} \nvDash \Diamond\mathsf{R}(x)$.

▶ **Example 15** (Merging)**.** In QLTL we can express the merging of two individuals at some point in the future with the predicate $\mathbf{willMerge}(x, y) := x \neq y \wedge \Diamond(x = y)$. In our running example in Figure 2, we have that in the first world $\sigma, \{x \mapsto a_0, y \mapsto c_0\} \vDash \mathbf{willMerge}(x, y)$, but clearly $\sigma, \{x \mapsto c_0, y \mapsto d_0\} \nvDash \mathbf{willMerge}(x, y)$.

▶ Remark 16 (Quantifier elision for unbound variables). A relevant difference with standard quantified logics is that, in QLTL, we cannot elide quantifications where the variable introduced does not appear in the subformula: taking any ϕ with $x \notin \mathrm{fv}(\phi)$ we have that in general $\exists x.\phi \not\equiv \phi$ and, similarly, $\forall x.\phi \not\equiv \phi$. More precisely, since we have that each $D(\omega)$ is not empty, the equivalences hold whenever ϕ does not contain any temporal operator.

We report here a concrete example: consider a world ω with a single individual $D(\omega) = \{s\}$ and a single looping counterpart function $\mathcal{C}\langle\omega, \omega\rangle = \{C\}$, where $C = \emptyset$ is the empty counterpart function. The trace is given by $\sigma = (C, C, \dots)$. By taking the empty assignment $\{\,\}$ and the closed formula $\phi = \mathsf{O}(\mathsf{tt})$, one can easily check that $\sigma, \{\,\} \vDash \mathsf{O}(\mathsf{tt})$, but $\sigma, \{\,\} \nvDash \exists x.\mathsf{O}(\mathsf{tt})$. The reason is that, once an assignment is extended with some element, stepping from one world to the next one requires every individual of the assignment to be transported and have a counterpart in the next world.

Alternatively, we could have restricted assignments in the semantics so that counterparts are required only for the free variables occurring in the formula. For example, the definition for the *next* operator $\mathsf{O}\phi$ becomes:

- $\sigma, \mu \vDash \mathsf{O}\phi$ *if* $C_0 \circ \mu_{|\mathrm{fv}(\phi)}$ *is defined and* $\sigma_1, C_0 \circ \mu_{|\mathrm{fv}(\phi)} \vDash \phi$.

For ease of presentation both in this paper and with respect to our Agda implementation, we consider the case where all elements in the context must have a counterpart. All our developments could anyhow be rephrased with the alternative approach.

▶ Remark 17 (No self-duality for *next*). We observe that, contrary to classical LTL, the next operator $\mathsf{O}\phi$ in our counterpart-style semantics in general is not self-dual with respect to negation, i.e.: $\neg\mathsf{O}\phi \not\equiv \mathsf{O}\neg\phi$. It is necessary to introduce a separate next operator that allows us to adequately capture the notion of negation. Notice that this absence of duality is due to the fact that we use partial functions in our counterpart model, which forces us to talk about the existence and the possible *absence* of a counterpart.

Consider again the counterpart model in Figure 2: it is easy to see that $\sigma_1, \{x \mapsto c_1\} \vDash \neg\mathsf{O}(\mathsf{B}(x))$, but $\sigma_1, \{x \mapsto c_1\} \not\vDash \mathsf{O}(\neg\mathsf{B}(x))$ since no counterpart for c_1 exists after one step.

The idea is that the *next* operator $\mathsf{O}\phi$ requires that a counterpart at the next step exists, and this will force us in Section 4 to present a dual next operator to deal with the absence of a counterpart. Indeed, the usual equivalence presented in LTL can be obtained again by restricting our models whose counterpart relations are functions which are total on their domain; this allows us to consider a unique trace of always-defined counterpart individuals, which in turn brings us back to a more standard LTL notion of trace.

3.2 Syntax and semantics of $\widehat{\mathsf{QLTL}}$

We can further extend the syntax of QLTL with an additional operator *weak until* $\phi_1 \mathsf{W} \phi_2$, as usually presented in LTL.

▶ **Definition 18** ($\widehat{\mathsf{QLTL}}$). *Let $\mathcal{X}$ be a set of variables and $x, y \in \mathcal{X}$. The set $\mathcal{F}^{\widehat{\mathsf{QLTL}}}$ of $\widehat{\mathsf{QLTL}}$ formulae is defined by the following rules:*

$$\psi = \mathsf{tt} \mid x = y \mid P(x), \qquad \phi := \psi \mid \neg\phi \mid \phi \vee \phi \mid \exists x.\phi \mid \mathsf{O}\psi \mid \phi_1 \mathsf{U} \phi_2 \mid \phi_1 \mathsf{W} \phi_2.$$

Since $\widehat{\mathsf{QLTL}}$ is an extension of QLTL, the semantics of common formulae is the same as the one in Definition 12, and we only present the semantics for the new operator *weak until*.

▶ **Definition 19** (Semantics of *weak until*). *The* weak until *operator is defined as follows:*

- $\sigma, \mu \vDash \phi_1 \mathsf{W} \phi_2$ *if at least one of the following conditions applies:*
  1. *either the same conditions for $\phi_1 \mathsf{U} \phi_2$ hold;*
  2. *for any $i \geq 0$ we have that $C_{\leq i} \circ \mu$ is defined and $\sigma_i, C_{\leq i} \circ \mu \vDash \phi_1$.*

▶ Remark 20 (*Always* operator $\Box\phi$). Similarly as with Remark 14, having the *weak until* operator allows us to directly define an *always* operator $\Box\phi := \phi \mathsf{W} \mathsf{ff}$. Equivalently, we explicitly provide its semantics as follows:

$\sigma, \mu \vDash \Box\phi$ *if for any $i \geq 0$ we have that $C_{\leq i} \circ \mu$ is defined and $\sigma_i, C_{\leq i} \circ \mu \vDash \phi$.*

For example, we have in Figure 2 that $\sigma, \{x \mapsto c_0\} \vDash \Diamond\Box\mathsf{R}(x)$ and $\sigma_2, \{x \mapsto c_2\} \vDash \Box\mathsf{B}(x)$. However, $\sigma, \{x \mapsto d_0\} \not\vDash \Box\mathsf{B}(x)$ since a counterpart is always required to exist.

▶ Remark 21 ($\widehat{\mathsf{QLTL}}$ is strictly more expressive than QLTL). It seems reasonable to also define the standard *always* operator in QLTL with $\Box\phi := \neg\Diamond\neg\phi$; however, this definition does *not* align with the semantics provided in Remark 20 for $\widehat{\mathsf{QLTL}}$, and it turns out that the $\Box\phi$ operator is *not* expressible in QLTL. By Theorem 41, it similarly can be seen that the *weak until* $\phi_1 \mathsf{W} \phi_2$ operator cannot be expressed in QLTL. This is due to the fact we are working in the setting of partial functions, and we will formally explain and present an intuition for this when we introduce the syntax and semantics of QLTL in PNF in Section 4.

As usual, two formulae are said to be equivalent whenever they are satisfied by the same traces and assignments.

▶ **Definition 22** ($\widehat{\mathsf{QLTL}}$ equivalence). *Given two $\widehat{QLTL}$ formulae $[\Gamma]\phi_1, [\Gamma]\phi_2$, defined on the same context Γ, we define equivalence as follows:*

$$\phi_1 \equiv \phi_2 \overset{def}{\iff} \forall \sigma, \mu \in \mathcal{A}^{\Gamma}_{\sigma_\bullet}.(\sigma, \mu \vDash \phi_1 \iff \sigma, \mu \vDash \phi_2).$$

Observe that the assignments we quantify on are defined on the same context Γ as the two formulae, and must again assign variables to the first world $\sigma_\bullet$ of the trace.

▶ **Theorem 23** (Equivalences in $\widehat{\mathsf{QLTL}}$). (`QLTL.Equivalences`) *As in LTL, the following equivalences also hold in $\widehat{QLTL}$:*

$$\phi_1 \mathsf{U} \phi_2 \equiv \phi_1 \mathsf{W} \phi_2 \wedge \Diamond \phi_2, \quad \phi_1 \mathsf{W} \phi_2 \equiv \phi_1 \mathsf{U} \phi_2 \vee \Box \phi_1.$$

▶ **Theorem 24** (Expansion laws in $\widehat{\mathsf{QLTL}}$). (`QLTL.ExpansionLaws`) *We have the following expansion laws in $\widehat{QLTL}$:*

$$\Diamond\phi \equiv \phi \vee \mathsf{O}(\Diamond\phi) \quad \phi_1 \mathsf{U} \phi_2 \equiv \phi_2 \vee (\phi_1 \wedge \mathsf{O}(\phi_1 \mathsf{U} \phi_2))$$
$$\Box\phi \equiv \phi \wedge \mathsf{O}(\Box\phi) \quad \phi_1 \mathsf{W} \phi_2 \equiv \phi_2 \vee (\phi_1 \wedge \mathsf{O}(\phi_1 \mathsf{W} \phi_2))$$

4 Positive Normal Form

Positive normal forms are a standard presentation of temporal logics, and can be used to simplify constructions and algorithms on both the theoretical and pratical implementation side [14, 3] while still preserving the full expressiveness of the original logic. The main purpose of this section is to provide the positive normal forms for both QLTL and $\widehat{\mathsf{QLTL}}$.

4.1 Positive normal form for QLTL

As observed in Remark 17, to present the positive normal form we need additional operators to adequately capture the negation of the temporal operators we described. Thus, we introduce a new flavour of the next operator, called *next-forall* $\mathsf{A}\phi$. Similarly, we have to introduce a negative dual to the *until* operator which we indicate as the *then* $\phi_1 \mathsf{T} \phi_2$ operator.

▶ **Definition 25** (QLTL in PNF). *Let $\mathcal{X}$ be a set of variables and $x, y \in X$. The set $\mathcal{F}^{PNF}$ of formulae of QLTL in **positive normal form** is generated by the following rules:*

$$\psi := \mathsf{tt} \mid x = y \mid P(x),$$

$$\phi := \psi \mid \neg\psi \mid \phi \vee \phi \mid \phi \wedge \phi \mid \exists x.\phi \mid \forall x.\phi \mid \mathsf{O}\phi \mid \mathsf{A}\phi \mid \phi_1 \mathsf{U} \phi_2 \mid \phi_1 \mathsf{T} \phi_2.$$

We now provide a satisfiability relation for PNF formulae by specifying the semantics just for the additional operators, omitting the ones that do not change.

▶ **Definition 26** (QLTL in PNF satisfiability). *We define the following additional constructs:*

- $\sigma, \mu \vDash \neg\psi$ *if* $\sigma, \mu \nvDash \psi$;
- $\sigma, \mu \vDash \phi_1 \wedge \phi_2$ *if* $\sigma, \mu \vDash \phi_1$ *and* $\sigma, \mu \vDash \phi_2$;
- $\sigma, \mu \vDash \forall x.\phi$ *if for any individual* $s \in D(\sigma_\bullet)$ *we have that* $\sigma, \mu[x \mapsto s] \vDash \phi$;
- $\sigma, \mu \vDash \mathsf{A}\phi$ *if whenever* $C_0 \circ \mu$ *is defined then* $\sigma_1, C_0 \circ \mu \vDash \phi$;
- $\sigma, \mu \vDash \phi_1 \mathsf{T} \phi_2$ *if at least one of the two conditions holds:*

- *there exists an* $\bar{n} \geq 0$ *such that, for any* $i < \bar{n}$*, if* $C_{\leq i} \circ \mu$ *is defined then* $\sigma_i, C_{\leq i} \circ \mu \vDash \phi_1$*; furthermore, if* $C_{\leq \bar{n}} \circ \mu$ *is defined then* $\sigma_{\bar{n}}, C_{\leq \bar{n}} \circ \mu \vDash \phi_2$*.*
- *for any* i*, if* $C_{\leq i} \circ \mu$ *is defined then* $\sigma_i, C_{\leq i} \circ \mu \vDash \phi_1$*.*

The intuition for the *next-forall* $\mathsf{A}\phi$ operator is that it allows us to capture the case where the counterpart of an individual does not exist at the next step: if it does, it is required to satisfy the formula ϕ.

Similarly to the *until* $\phi_1 \mathsf{U} \phi_2$ operator, the *then* $\phi_1 \mathsf{T} \phi_2$ operator allows us to take a sequence of worlds where ϕ_1 is satisfied for some steps until ϕ_2 holds. The crucial observation is that both the intermediate counterparts satisfying ϕ_1 and the conclusive counterpart satisfying ϕ_2 are allowed to not exist, and indeed any trace consisting of all empty counterpart functions always satisfies $\phi_1 \mathsf{T} \phi_2$. As a special case, $\phi_1 \mathsf{T} \phi_2$ can also be validated by considering a trace in which every counterpart of the initial assignment exists, and they all satisfy ϕ_1.

▶ Remark 27 (Alternative definition of *then*). It could be wondered whether the *then* $\phi_1 \mathsf{T} \phi_2$ operator can be expressed by requiring that counterparts *exist* where ϕ_1 is satisfied, and allow non-existence just for the assignment that satisfies ϕ_2. It turns out that this variation is equivalent to the one shown here, and we have formally shown this equivalence in Agda (`Alternative.QLTL`). We present the current one as it lends itself to a more intuitive representation of the expansion law shown in Theorem 41.

▶ **Example 28** (*Then and next-forall*)**.** In our running example in Figure 2, we illustrate the possibility for $\mathsf{B}(x)\mathsf{T}\mathsf{R}(x)$ and $\mathsf{A}\mathsf{B}(x)$ to be satisfied even when a counterpart does not exist after one or more steps. In particular, it can be verified that $\sigma, \{x \mapsto c_0\} \vDash \mathsf{B}(x)\mathsf{T}\mathsf{R}(x)$ holds since $\mathsf{R}(x)$ is eventually satisfied while $\mathsf{B}(x)$ holds, just like the *until* operator. We have that both $\sigma, \{x \mapsto a_0\} \vDash \mathsf{A}\mathsf{R}(x)$ and $\sigma_1, \{x \mapsto c_1\} \vDash \mathsf{A}\mathsf{R}(x)$ hold, since no counterpart for c_1 exists after one step. Finally, we have that $\sigma, \{x \mapsto d_0\} \vDash \mathsf{B}(x)\mathsf{T}\mathsf{R}(x)$ holds since $\mathsf{B}(x)$ holds but no counterpart exists after two steps, and $\sigma_2, \{x \mapsto c_2\} \vDash \mathsf{B}(x)\mathsf{T}\mathsf{R}(x)$ since a counterpart always exists but $\mathsf{B}(x)$ holds forever.

The crucial observation that validates the PNF presented in Section 4 is that the negation of both the *next* $\mathsf{O}\phi$ and *until* $\phi_1 \mathsf{U} \phi_2$ formulae can now be expressed inside the logic. We will explicitly indicate with $\vDash_{\mathsf{QLTL}}$ and $\vDash_{\mathsf{PNF}}$ the satisfiability relations defined for formulae in standard QLTL and QLTL in PNF, respectively.

▶ **Theorem 29** (Negation of *next* and *until* is expressible in PNF)**.** (`QLTL.Negation`)
Let ψ *be an atomic formula in PNF. Then we have*

$$\forall \sigma, \mu \in \mathcal{A}^{\Gamma}_{\sigma_\bullet}. \sigma, \mu \vDash_{QLTL} \neg \mathsf{O}(\psi) \iff \sigma, \mu \vDash_{PNF} \mathsf{A}(\neg \psi)$$

$$\forall \sigma, \mu \in \mathcal{A}^{\Gamma}_{\sigma_\bullet}. \sigma, \mu \vDash_{QLTL} \neg (\psi_1 \mathsf{U} \psi_2) \iff \sigma, \mu \vDash_{PNF} (\neg \psi_2) \mathsf{T} (\neg \psi_1 \wedge \neg \psi_2).$$

Similarly, we have that the negation of the new operators is itself contained in PNF.

▶ **Proposition 30** (Negation of *then* and *next-forall* is in PNF)**.** (`PNF.Negation`)
Let ψ *be an atomic formula in PNF. Then we have*

$$\forall \sigma, \mu \in \mathcal{A}^{\Gamma}_{\sigma_\bullet}. \sigma, \mu \nvDash_{PNF} \mathsf{A}(\psi) \iff \sigma, \mu \vDash_{PNF} \mathsf{O}(\neg \psi)$$

$$\forall \sigma, \mu \in \mathcal{A}^{\Gamma}_{\sigma_\bullet}. \sigma, \mu \nvDash_{PNF} \psi_1 \mathsf{T} \psi_2 \iff \sigma, \mu \vDash_{PNF} (\neg \psi_2) \mathsf{U} (\neg \psi_1 \wedge \neg \psi_2).$$

Notice how the previous results can be easily generalized to the case where full formulae ϕ are considered. These equivalences allow us to define a formal translation $\overline{\,\cdot\,} : \mathcal{F}^{\mathsf{QLTL}} \to \mathcal{F}^{\mathsf{PNF}}$

from the QLTL syntax presented in Definition 12 to the current one in PNF, preserving equivalence of formulae. This is done with the obvious syntactical transformation that pushes the negation in QLTL formulae down to elementary formulae and replaces temporal operators with their negated counterpart. For example:

$$\overline{\mathsf{O}\phi} := \mathsf{O}\overline{\phi} \qquad \overline{\neg\mathsf{O}\phi} := \mathsf{A}\overline{\neg\phi} \qquad \overline{\phi_1\mathsf{U}\phi_2} := \overline{\phi_1}\mathsf{U}\overline{\phi_2} \qquad \overline{\neg\phi_1\mathsf{U}\phi_2} := \overline{\neg\phi_2}\mathsf{T}(\overline{\neg\phi_1} \wedge \overline{\neg\phi_2})$$

▶ **Theorem 31** (PNF equivalence). (PNF.Conversion) *Let $\overline{\cdot}$: $\mathcal{F}^{QLTL} \to \mathcal{F}^{PNF}$ be the aforementioned syntactical translation that replaces negated temporal operators with their equivalent ones in PNF. For any QLTL formula $[\Gamma]\phi \in \mathcal{F}^{QLTL}$, we can express the following result:*

$$\forall \sigma, \mu \in \mathcal{A}^{\Gamma}_{\sigma_\bullet}.\sigma, \mu \vDash_{QLTL} \phi \iff \sigma, \mu \vDash_{PNF} \overline{\phi}.$$

▶ Remark 32 (*Weak until* and *then*). In classic LTL, the *weak until* operator $\phi_1\mathsf{W}\phi_2$ and *until* operator $\phi_1\mathsf{U}\phi_2$ can each be used to express the negation of the other; in QLTL, it is easy to see that the *then* operator $\phi_1\mathsf{T}\phi_2$ does *not* provide the same semantics of the *weak until* definition $\phi_1\mathsf{W}\phi_2$ as we presented it in Definition 19.

The crucial difference between the two operators is in the deallocation of elements: the *then* $\phi_1\mathsf{T}\phi_2$ operator holds even if no counterpart for the current element exists, while the *weak until* $\phi_1\mathsf{W}\phi_2$ operator requires for a counterpart to exist at every step. Since the temporal operators of QLTL ensure that counterparts exist, their dual operators must be able to show that *no* counterpart can exist while still satisfying the formula. Indeed, the absence of counterparts can be expressed with the $\mathsf{A}\phi$ and $\phi_1\mathsf{T}\phi_2$ operators.

▶ Remark 33 (Non-expressibility of $\Box\phi$ and $\phi_1\mathsf{W}\phi_2$ in QLTL). As previously mentioned in Remark 21, the $\Box\phi$ and $\phi_1\mathsf{W}\phi_2$ operators are not expressible in standard QLTL, and need to be defined as primitives in an extended logic $\widehat{\mathsf{QLTL}}$.

The reason is made clearer now that we have explicitly seen how the negations of the *until* and *next* operators behave. The idea for this restriction in expressive power is that the operators considered in QLTL cannot be manipulated into providing the semantics of the always operator $\Box\phi$: the *next* $\mathsf{O}\phi$ and *until* $\phi_1\mathsf{U}\phi_2$ operators only consider finite prefixes of a trace; on the other hand, their corresponding negations *next-forall* $\mathsf{A}\phi$ and *then* $\phi_1\mathsf{T}\phi_2$ operators allow for counterparts not to exist.

4.2 Positive normal form for $\widehat{\mathsf{QLTL}}$

In this section we extend the PNF introduced for QLTL to the case of $\widehat{\mathsf{QLTL}}$ and consider the additional operators required to express it.

It turns out that to obtain the positive normal for $\widehat{\mathsf{QLTL}}$ form we need to define yet another operator *until-forall* $\phi_1\mathsf{F}\phi_2$ that expresses the negation of the *weak-until* $\phi_1\mathsf{W}\phi_2$ operator. This fully completes the picture of the temporal operators required.

▶ **Definition 34** ($\widehat{\mathsf{QLTL}}$ in PNF). *Let $\mathcal{X}$ be a set of variables with $x, y \in \mathcal{X}$. The set $\mathcal{F}^{\widehat{PNF}}$ of formulae of $\widehat{\mathsf{QLTL}}$ in **positive normal form** is generated by the following rules:*

$$\psi := \mathsf{tt} \mid x = y \mid P(x),$$

$$\phi := \psi \mid \neg\psi \mid \phi \vee \phi \mid \phi \wedge \phi \mid \exists x.\phi \mid \forall x.\phi \mid \mathsf{O}\phi \mid \mathsf{A}\phi \mid \phi_1\mathsf{U}\phi_2 \mid \phi_1\mathsf{T}\phi_2 \mid \phi_1\mathsf{W}\phi_2 \mid \phi_1\mathsf{F}\phi_2.$$

▶ **Definition 35** (Semantics of *until-forall*). *The* until-forall *operator is defined as follows:*

- $\sigma, \mu \vDash \phi_1\mathsf{F}\phi_2$ *if there exists an $\bar{n} \geq 0$ such that:*

- *for any $i < \bar{n}$, if $C_{\leq i} \circ \mu$ is defined then $\sigma_i, C_{\leq i} \circ \mu \vDash \phi_1$;*
- *if $C_{\leq \bar{n}} \circ \mu$ is defined then $\sigma_{\bar{n}}, C_{\leq \bar{n}} \circ \mu \vDash \phi_2$.*

The intuition for the *until-forall* $\phi_1 \mathsf{F} \phi_2$ is similar to the classical *until* $\phi_1 \mathsf{U} \phi_2$, with the caveat that the counterparts are allowed not to exist. Crucially, we observe how the definition of the *until-forall* $\phi_1 \mathsf{F} \phi_2$ operator coincides with the first satisfiability condition of the *then* $\phi_1 \mathsf{T} \phi_2$ operator presented in Definition 26.

Now that we have defined the complete set of the temporal operators, the second *then* condition can similarly be expressed by a derived *always-forall* $\Box^* \phi$ operator, which we present along with a *eventually-forall* $\Diamond^* \phi$ operator.

Similarly as with the *then* and *until-forall* operators, the difference with their standard counterparts *eventually* $\Diamond \phi$ and *always* $\Box \phi$ is that they can still be satisfied even when counterparts do not exist.

▶ Remark 36 (*Always-forall* and *eventually-forall*). The *always-forall* and *eventually-forall* operators are defined as $\Box^* \phi := \phi \mathsf{T} \mathsf{ff}$ and $\Diamond^* \phi := \mathsf{tt} \mathsf{F} \phi$, respectively. Their semantics can be explicitly presented as follows:

- $\sigma, \mu \vDash \Box^* \phi$ *for any i, if $C_{\leq i} \circ \mu$ is defined then $\sigma_i, C_{\leq i} \circ \mu \vDash \phi$;*
- $\sigma, \mu \vDash \Diamond^* \phi$ *if there exists $i \geq 0$ such that if $C_{\leq i} \circ \mu$ is defined then $\sigma_i, C_{\leq i} \circ \mu \vDash \phi$.*

The equivalence between $\widehat{\mathsf{QLTL}}$ and its positive normal form can again be retrieved again by observing the following equivalences:

▶ **Theorem 37** (Negation of *weak-until* and *until-forall* is expressible in PNF). (QLTL.Ext.Negation) *Let ψ be an atomic formula. Then we have*

$$\forall \sigma, \mu \in \mathcal{A}^{\Gamma}_{\sigma_\bullet}. \sigma, \mu \vDash_{\widehat{QLTL}} \neg(\psi_1 \mathsf{W} \psi_2) \iff \sigma, \mu \vDash_{\widehat{PNF}} (\neg \psi_2) \mathsf{F} (\neg \psi_1 \wedge \neg \psi_2),$$

$$\forall \sigma, \mu \in \mathcal{A}^{\Gamma}_{\sigma_\bullet}. \sigma, \mu \nvDash_{\widehat{PNF}} \psi_1 \mathsf{F} \psi_2 \iff \sigma, \mu \vDash_{\widehat{PNF}} (\neg \psi_2) \mathsf{W} (\neg \psi_1 \wedge \neg \psi_2).$$

▶ **Proposition 38** (Negation for *always-forall* and *eventually-forall*). (All.Negation) *Let ψ be an atomic formula. Then we have*

$$\forall \sigma, \mu \in \mathcal{A}^{\Gamma}_{\sigma_\bullet}. \sigma, \mu \vDash_{\widehat{QLTL}} \neg \Diamond \psi \iff \sigma, \mu \vDash_{\widehat{PNF}} \Box^* (\neg \psi),$$

$$\forall \sigma, \mu \in \mathcal{A}^{\Gamma}_{\sigma_\bullet}. \sigma, \mu \vDash_{\widehat{QLTL}} \neg \Box \psi \iff \sigma, \mu \vDash_{\widehat{PNF}} \Diamond^* (\neg \psi),$$

$$\forall \sigma, \mu \in \mathcal{A}^{\Gamma}_{\sigma_\bullet}. \sigma, \mu \nvDash_{\widehat{PNF}} \Diamond^* \psi \iff \sigma, \mu \vDash_{\widehat{PNF}} \Box (\neg \psi),$$

$$\forall \sigma, \mu \in \mathcal{A}^{\Gamma}_{\sigma_\bullet}. \sigma, \mu \nvDash_{\widehat{PNF}} \Box^* \psi \iff \sigma, \mu \vDash_{\widehat{PNF}} \Diamond (\neg \psi).$$

We can extend the formal translation $\overline{\cdot} : \mathcal{F}^{\widehat{\mathsf{QLTL}}} \to \mathcal{F}^{\widehat{\mathsf{PNF}}}$ to deal with the new operator:

$$\overline{\phi_1 \mathsf{W} \phi_2} := \overline{\phi_1} \mathsf{W} \overline{\phi_2} \qquad \overline{\neg(\phi_1 \mathsf{W} \phi_2)} := \overline{\neg \phi_2} \mathsf{F} (\overline{\neg \phi_1} \wedge \overline{\neg \phi_2})$$

As with Theorem 31, we obtain a similar result for the case of $\widehat{\mathsf{QLTL}}$:

▶ **Theorem 39** ($\widehat{\mathsf{PNF}}$ equivalence). (PNF.Conversion) *For any $\widehat{QLTL}$ formula $[\Gamma]\phi \in \mathcal{F}^{\widehat{QLTL}}$ we have that:*

$$\forall \sigma, \mu \in \mathcal{A}^{\Gamma}_{\sigma_\bullet}. \sigma, \mu \vDash_{\widehat{QLTL}} \phi \iff \sigma, \mu \vDash_{\widehat{PNF}} \overline{\phi}.$$

Each pair of operators can now be uniformly expressed in terms of each other.

▶ **Theorem 40** (Equivalences between operators in $\widehat{\mathsf{PNF}}$). (All.Equivalences) *The following equivalences hold in $\widehat{\mathsf{PNF}}$:*

$$\phi_1 \mathsf{U} \phi_2 \equiv \phi_1 \mathsf{W} \phi_2 \wedge \Diamond \phi_2 \quad \phi_1 \mathsf{W} \phi_2 \equiv \phi_1 \mathsf{U} \phi_2 \vee \Box \phi_1$$
$$\phi_1 \mathsf{F} \phi_2 \equiv \phi_1 \mathsf{T} \phi_2 \wedge \Diamond^* \phi_2 \quad \phi_1 \mathsf{T} \phi_2 \equiv \phi_1 \mathsf{F} \phi_2 \vee \Box^* \phi_1.$$

As shown by the following result, each operator introduced so far can be described with an expansion law that allows for the operator to be specified in terms of itself, recursively.

▶ **Theorem 41** (Expansion laws). (All.ExpansionLaws) *The following hold in $\widehat{\mathsf{PNF}}$:*

$$\phi_1 \mathsf{U} \phi_2 \equiv \phi_2 \vee (\phi_1 \wedge \mathsf{O}(\phi_1 \mathsf{U} \phi_2)) \qquad \phi_1 \mathsf{T} \phi_2 \equiv \phi_2 \vee (\phi_1 \wedge \mathsf{A}(\phi_1 \mathsf{T} \phi_2))$$
$$\phi_1 \mathsf{W} \phi_2 \equiv \phi_2 \vee (\phi_1 \wedge \mathsf{O}(\phi_1 \mathsf{W} \phi_2)) \qquad \phi_1 \mathsf{F} \phi_2 \equiv \phi_2 \vee (\phi_1 \wedge \mathsf{A}(\phi_1 \mathsf{F} \phi_2)).$$

The *then* $\phi_1 \mathsf{T} \phi_2$ and *until-forall* $\phi_1 \mathsf{F} \phi_2$ operators satisfy a similar expansion law as the *until* $\phi_1 \mathsf{T} \phi_2$ and *weak until* operators $\phi_1 \mathsf{W} \phi_2$, but by dually employing the *next-forall* $\mathsf{A}\phi$ operator to advance in the sequence.

▶ Remark 42 (Functional counterparts collapse the semantics). As briefly mentioned in Remark 17, when our counterpart model is restricted to functions that are always defined and are total on their domain we actually have that the pairs of operators previously introduced collapse and provide the same semantics of the classical operators. In particular, we obtain that $\mathsf{O}\phi \equiv \mathsf{A}\phi$, $\phi_1 \mathsf{U} \phi_2 \equiv \phi_1 \mathsf{F} \phi_2$, $\phi_1 \mathsf{W} \phi_2 \equiv \phi_1 \mathsf{T} \phi_2$, and this fact in turn allows us to obtain the same semantics and dualities of the classic presentation of LTL.

▶ Remark 43 (Temporal operators as fixpoints). We remark how in a set-based semantics with fixpoints the *until-forall* $\phi_1 \mathsf{F} \phi_2$ and *then* $\phi_1 \mathsf{T} \phi_2$ correspond to, respectively, least and greatest fixpoints of the expansion law presented in Theorem 41.

5 Conclusions and future works

In this paper we presented a positive normal form for quantified temporal logics based on counterpart semantics. We have shown how counterpart-based models provide a solution to the trans-world problem, and allow for an adequate modelling of deallocation and merging for the components of a system even from a practical point of view. To ensure a proper treatment of negation, we introduced and investigated a positive normal form that can explicitly deal with the notion of deallocation and it is pivotal for the use of fixpoints.

Several issues arise from the identification of a suitable positive normal form, leading to a variety of different perspectives for future work.

The flexibility of our approach, combined with the minimality of our syntax, suggests that extending our results to different or richer logics should be straightforward. For example, we believe that our approach can be used to obtain a counterpart semantics and a positive normal form presentation for CTL [8] and a second-order version of LTL.

From a semantical point of view there are at least two possible directions that can be explored, through the generalization or specialization of the models we introduce.

In particular, the counterpart models we considered only allow for the counterpart relation to be functional; this choice aims to maintain a tight correspondence with the original context of graph rewriting, where these modal counterpart logics were presented [5, 6, 7]. We believe that our approach can be easily extended in the setting of arbitrary counterpart relations, which allow for the duplication of individuals. This, however, requires a more precise analysis for the case of fixpoints and how they can be used to derive the usual temporal operators.

Finally, specializing the worlds of the models to be algebras instead of plain sets, as it is presented in [10] and [11], can be easily done while still preserving the positive normal form results we obtained.

References

1 F. Belardinelli. *Quantified Modal Logic and the Ontology of Physical Objects*. PhD thesis, Scuola Normale Superiore of Pisa, 2006.

2 P. Blackburn, J.F.A.K. van Benthem, and F. Wolter, editors. *Handbook of Modal Logic*, volume 3. North-Holland, 2007.

3 D. Bustan, A. Flaisher, O. Grumberg, O. Kupferman, and M.Y. Vardi. Regular vacuity. In D. Borrione and W. J. Paul, editors, *CHARME 2005*, volume 3725 of *LNCS*, pages 191–206. Springer, 2005.

4 L. Cardelli, P. Gardner, and G. Ghelli. A spatial logic for querying graphs. In P. Widmayer, S. Eidenbenz, F. Triguero, R. Morales, R. Conejo, and M. Hennessy, editors, *ICALP 2002*, pages 597–610. Springer, 2002.

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

6 B. Courcelle. The expression of graph properties and graph transformations in monadic second-order logic. In G. Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations*, pages 313–400. World Scientific, 1997.

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

8 E. Allen Emerson. Temporal and modal logic. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*, pages 995–1072. Elsevier and MIT Press, 1990.

9 E. Franconi and D. Toman. Fixpoint extensions of temporal description logics. In D. Calvanese, G. De Giacomo, and E. Franconi, editors, *DL 2003*, volume 81 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2003.

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

11 F. Gadducci and D. Trotta. A presheaf semantics for quantified temporal logics. *CoRR*, abs/2111.03855, 2021. URL: `https://arxiv.org/abs/2111.03855`, `arXiv:2111.03855`.

12 A. Hazen. Counterpart-theoretic semantics for modal logic. *The Journal of Philosophy*, 76(6):319–338, 1979.

13 I.M. Hodkinson, F. Wolter, and M. Zakharyaschev. Monodic fragments of first-order temporal logics: 2000-2001 A.D. In R. Nieuwenhuis and A. Voronkov, editors, *LPAR 2001*, volume 2250 of *LNCS*, pages 1–23. Springer, 2001.

14 S. Huang and R. Cleaveland. A tableau construction for finite linear-time temporal logic. *Journal of Logic and Algebraic Methods in Programming*, 125:100743, 2022.

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

16 U. Norell. Dependently typed programming in Agda. In A. Kennedy and A. Ahmed, editors, *TLDI 2009*, pages 1–2. ACM, 2009.

17 A. Pnueli. The temporal logic of programs. In *FOCS 1977*, pages 46–57. IEEE Computer Society, 1977.

18 J.H. Reif and A.P. Sistla. A multiprocess network logic with temporal and spatial modalities. *Journal of Computer and System Science*, 30(1):41–53, 1985.