A Sound and Complete Proof System for QPTL

TIM FRENCH AND MARK REYNOLDS

ABSTRACT. In this paper we give sound and complete proof systems for the useful and expressive *quantified propositional temporal logic*, both with and without past temporal operators. Until now an axiomatization has only existed for the version with the inclusion of the past operators and this axiomatization relied on the past operators in subtle and complicated ways. In certain situations, such as in branching time extensions of the linear time logic, it is important to avoid using the past time operators. Our completeness proof proceeds mostly by using deterministic Rabin automata but the main step is via a new and interesting proof of correctness for an optimal complementation procedure for nondeterministic Büchi automata.

1 Introduction

Quantified propositional temporal logic, QPTL [16], extends traditional propositional temporal logic, PLTL [12], and has the ability to quantify over interpretations of the propositional atoms. It is a logic used for reasoning about structures with atoms whose truth values vary along a discrete, linearly ordered flow of time. The set of formulas of QPTL is built up recursively from the atoms via classical negation and disjunction, temporal next-time and future-always connectives (from the standard PLTL) and quantification over atoms. We also have $\forall x \alpha$ being true at a moment of time in a structure if and only if all other structures which differ only in the interpretation of the atom x make α true at that moment.

Standard PLTL is often (see for example [19]) extended to a logic which we here call PLTLP (in view of a lack of standard terminology) by the addition of temporal operators which refer to the past sequence of states leading up to the point of evaluation (in addition to the usual future time operators). A complete axiomatization is given in [5] (see also [6]) for an extension of QPTL that involves these past-time operators. We shall call this extension QPTLP although it is called QPTL in [5]. The sketch of the completeness proof given in that paper shows that for every non-deterministic Büchi automaton, A, there is a formula of QPTLP, χ_A that is satisfied by exactly the ω-sequences that A accepts and that every formula of QPTLP

Advances in Modal Logic, Volume 4, 127–147.

is equivalent to a non-deterministic Büchi automaton. It should be noted that the proof makes essential use of the past-time operators and restricting the proof system to only the future axioms does not give a complete axiomatization for QPTL.

In our paper we produce a complete axiomatization for pure (future only) QPTL that does not rely on past operators. This is important in various circumstances. For example, when the linear time logic is extended to a branching time version then there are many complications with the semantics for the past time operators [2]. In any case, the past-time operators can, in a certain useful way, be easily defined in terms of the future only operators and quantification.

Our axiom system is a very simple extension of the traditional axiom system [4] for PLTL via a natural induction rule. The completeness proof is also considerably simpler (albeit a little too long to appear in full here). We use deterministic Rabin automata to show that every QPTL formula is provably equivalent to an existentially quantified formula (a Σ_1 formula). Completeness follows from the equivalence between satisfiability and consistency in PLTL and QPTL.

We also show that we obtain a sound and complete axiomatization for QPTLP by the simple addition of the standard PLTLP axioms for the past time operators. We consider the expressiveness of QPTLP versus QPTL: are the past operators important when we do not need them for our completeness proof?

There are important applications of QPTL in verification of complex systems because of the two main recognized advantages of quantification: extra expressivity and the ability to reason about refinement relations between programs [9]. There are well known inadequacies in the expressiveness of PLTL [17] and it is also well known [15] that QPTL is one of several ways of achieving an improvement to reach the important standard of expressiveness of automata and the second order monadic logic of one successor (S1S). If one program produces the same observable behaviour as another but perhaps uses extra variables with perhaps some extra computation steps in between the observed steps then we say that the second program *refines* or *implements* the first. As shown in [10], refinement can be defined via the existence of temporal propositions which satisfy certain temporal properties, i.e. via a formula of QPTLP. There are also important applications of propositional quantification in work on logics of knowledge [1] where we want to quantify over propositions which are local to a particular agent in the sense that their truth value only depends on the state of the agent.

We define the logics in the next section before presenting our proof system. In section 5 we introduce automata and in section 6 we show that every QPTL formula is provably equivalent to a Σ_1 formula. The next section sketches the remainder of the completeness proof: it will appear in full in a longer version of this paper later. We conclude with some discussion of extensions of the results.

2 Syntax and semantics

The language QPTL consists of an infinite set of atomic variables $\mathcal{V} = \{x_0, x_1, \ldots\}$, the boolean operations $\neg, \vee$, (**not** and **or** respectively) and the future temporal operators $\bigcirc, \Box$ (**next** and **generally** respectively), along with the quantifier $\forall$ (**for all**). Later in the paper we will also consider an extended version of the language, QPTLP that also includes the past temporal operators $\circleddash, \boxminus$ (**weak previous** and **past generally** respectively). We will use the convention that all unary operators (including $\forall$) bind weaker than binary operators, unless otherwise indicated by brackets. The set of formulas of QPTL contains x, for all $x \in \mathcal{V}$ and is closed under the usual temporal operators $\neg$, $\vee$, $\bigcirc$, and $\Box$ as well as containing $\forall x \alpha$ for all $x \in \mathcal{V}$.

We will next present the definition of a structure σ that evaluates a formula in the language QPTL. A structure $\sigma = (\sigma_0, \sigma_1, \ldots)$ is a countably infinite sequence of elements of $\wp(\mathcal{V})$, (for each $i \in \mathbb{N}$, $\sigma_i \subseteq \mathcal{V}$). We say $\sigma' = (\sigma'_0, \sigma'_1, \ldots)$ is an *X-variant* of σ for some finite $X \subset \mathcal{V}$, if $\forall i\ \sigma_i \backslash X = \sigma'_i \backslash X$. For the singleton $\{x\}$, we refer to an x-variant, rather than an $\{x\}$-variant. Given some structure σ we define the j^{th} moment of σ be the tuple (σ, j) for $j \in \mathbb{N}$ and inductively define a formula α to "hold" at the j^{th} moment of σ (denoted $(\sigma, j) \models \alpha$) as follows.

$(\sigma, j) \models x \Leftrightarrow x \in \sigma_j$, for all $x \in \mathcal{V}$. $\quad (\sigma, j) \models \neg\alpha \Leftrightarrow (\sigma, j) \not\models \alpha$.
$(\sigma, j) \models \alpha_1 \vee \alpha_2 \Leftrightarrow$ either $(\sigma, j) \models \alpha_1$ or $(\sigma, j) \models \alpha_2$.
$(\sigma, j) \models \bigcirc\alpha \Leftrightarrow (\sigma, j+1) \models \alpha$. $\quad (\sigma, j) \models \Box\alpha \Leftrightarrow \forall k \geq j \quad (\sigma, k) \models \alpha$.
$(\sigma, j) \models \forall x \alpha \Leftrightarrow (\sigma', j) \models \alpha$, for all $x-$variants σ' of σ.

We say σ is an initial model of α (written $\sigma \models \alpha$) if $(\sigma, 0) \models \alpha$. If for every structure σ and for every $i \in \mathbb{N}$ we have $(\sigma, i) \models \alpha$, we say α is valid (written $\models \alpha$), and if $\neg\alpha$ is not valid, we say α is satisfiable.

We complete this section by defining additional operators in terms of those already defined. The propositional operators $\wedge, \rightarrow, \leftrightarrow$ are well known to be expressible in terms of $\neg$ and $\vee$. We will also consider the formulas $\top, \bot$ (respectively "true" and "false") to be abbreviations for, respectively $(x_0 \vee \neg x_0)$ and $\neg(x_0 \vee \neg x_0)$. Given the sets of propositions $X = \{x_0, \ldots, x_m\}$ and $Y = \{y_0, \ldots, y_m\}$ we use the abbreviations $\exists X$ and $X \leftrightarrow Y$ for $\exists x_0 \ldots \exists x_m$ and $\bigwedge_{i=0}^{m} x_i \leftrightarrow y_i$ respectively. For $i \in \mathbb{N}$, and any formula α we define $\bigcirc^i \alpha$ inductively by $\bigcirc^0 \alpha = \alpha$ and $\bigcirc^{i+1} \alpha = \bigcirc\bigcirc^i \alpha$. We will also use the abbreviation $\bigcirc^{\leq n}\alpha$ for $\bigwedge_{i=0}^{n}\bigcirc^{i}\alpha$. The other abbreviations in QPTL are:

$$
\begin{array}{ll}
\exists x\alpha = \neg\forall x\neg\alpha & \Diamond\alpha = \neg\Box\neg\alpha \\
\alpha\mathcal{W}\beta = \exists u(u \wedge \Box(u \rightarrow ((\alpha \wedge \bigcirc u) \vee \beta))) & \alpha\mathcal{U}\beta = \Diamond\beta \wedge (\alpha\mathcal{W}\beta)
\end{array}
$$

where $u = x_i$ for the least i such that $x_i$does not appear in α or β. These operators will be referred to as, respectively: existential quantification, future, waiting, and until. Intuitively, $\Diamond\alpha$ is true if α is true at some future moment, and $\alpha\mathcal{U}\beta$ is true if α is true for all moments until β is true.

The above syntax and semantics without quantification define a language we will refer to as PLTL. Full PLTL also includes the $\mathcal{U}$ operator, but this is not necessary in QPTL.

3 The proof system

In this section we will give the axioms and rules of inference for QPTL. We will also derive some basic theorems using the system, which will be used in the proof of completeness. We let $var(\alpha)$ be set of variables occurring in α. If α and β are formulas of QPTL, then we define the substitution $\alpha[\beta\backslash x]$ to be the formula α with all occurrences of x replaced by the sub-formula β. We say x is *free for* β in α if

1. α does not contain a sub-formula of the form $\forall x\gamma$ and

2. for all $y \in var(\beta)$, if α contains a sub-formula $\forall y\gamma$ then $x \notin var(\gamma)$.

We now give the axiom system for QPTL which we will denote **QF**. It is a direct extension of the axiom system **DX** which was given in [4] and was shown to be sound and complete for PLTL (without the $\mathcal{U}$ operator). The axioms are substitution instances of the following:

C0	any propositional tautology	C1	$\Box(\alpha \rightarrow \beta) \rightarrow (\Box\alpha \rightarrow \Box\beta)$
C2	$\Box\alpha \rightarrow (\alpha \wedge \bigcirc(\Box\alpha))$	C3	$\bigcirc\neg\alpha \leftrightarrow \neg\bigcirc\alpha$
C4	$\bigcirc(\alpha \rightarrow \beta) \rightarrow (\bigcirc\alpha \rightarrow \bigcirc\beta)$	C5	$\Box(\alpha \rightarrow \bigcirc\alpha) \rightarrow (\alpha \rightarrow \Box\alpha)$
QX0	$\exists x(x \wedge \bigcirc\Box\neg x)$	QX1	$\forall x\bigcirc\alpha \leftrightarrow \bigcirc\forall x\alpha$
QX2	$\forall x\alpha \rightarrow \alpha[\beta\backslash x]$	where	x is free for β in α.

There are also four rules of inference, Modus Ponens and Temporal Generalization (from DX), $\forall$-Generalization (from [5]) and *Quantified Induction*, which is new. The first three rules are given as:

$$
\text{MP}\ \frac{\alpha \rightarrow \beta,\ \alpha}{\beta} \qquad \text{T}-\text{Gen}\ \frac{\alpha}{\Box\alpha} \qquad \forall\text{–Gen}\ \frac{\alpha \rightarrow \beta}{\alpha \rightarrow \forall x\beta} \quad \text{when } x \notin var(\alpha).
$$

To state the Quantified Induction rule we require the following definitions. Let $X = \{x_0, \ldots, x_m\}$ and $Y = \{y_0, \ldots, y_m\}$ be disjoint sets of variables. We say a formula β has *temporal depth k in X* if every occurrence of $x \in X$ in β does not occur in the scope of a $\Box$ operator, or more than k $\bigcirc$ operators. The rule is:

$$\text{QI} \qquad \frac{\beta \to \exists \mathrm{Y}(\bigcirc^{\leq 1}(\mathrm{Y} \leftrightarrow \mathrm{X}) \wedge \bigcirc \beta[\mathrm{Y} \backslash \mathrm{X}])}{\beta \to \exists \mathrm{X} \Box \beta},$$

where β has temporal depth 1 in X and the variables of Y do not appear in β.

3.1 Some theorems

Using the rule $\forall$-Gen and the axiom QX2 we can generate the substitution rule. Since the axiom system **QF** directly extends the sound and complete axiom system **DX** [4], any validity of PLTL can be assumed to be provable in **QF**, and formulas of QPTL can be substituted for free variables. This will be referred to as *semantic validification (SV)*. In most cases we will leave the proof of semantic validity for such a formula to the reader, and denote such derivations as PLTL, in the proofs.

We also note that the axioms C0, QX2 and the rule $\forall$-Gen give a sound and complete axiom system for quantified propositional logic (QPL), so semantic validification also applies to any theorem not containing temporal operators (denoted as QPL in the proofs). We will now prove two theorems that involve both temporal operators and quantifiers. Note that we use $\vdash$ to mean provable in **QF**.

(1) $\Box(\alpha \vee \beta) \to \exists z \Box((\alpha \wedge z) \vee (\beta \wedge \neg z))$.

PLTL	$\vdash$	$(\alpha \vee \beta) \to ((\alpha \wedge \alpha) \vee (\beta \wedge \neg\alpha))$
T-Gen	$\vdash$	$\Box((\alpha \vee \beta) \to ((\alpha \wedge \alpha) \vee (\beta \wedge \neg\alpha)))$
C1	$\vdash$	$\Box(\alpha \vee \beta) \to \Box((\alpha \wedge \alpha) \vee (\beta \wedge \neg\alpha))$
QX2	$\vdash$	$\Box(\alpha \vee \beta) \to \exists z \Box((\alpha \wedge z) \vee (\beta \wedge \neg z))$

(2) $\vdash \Box \forall x \alpha \leftrightarrow \forall x \Box \alpha$.

		$(\longrightarrow)$			$(\longleftarrow)$
QX2	$\vdash$	$\forall x \alpha \to \alpha$	C2, SV	$\vdash$	$\forall x \Box \alpha \to \forall x \bigcirc \Box \alpha$
T-Gen	$\vdash$	$\Box(\forall x \alpha \to \alpha)$	QX1	$\vdash$	$\forall x \Box \alpha \to \bigcirc \forall x \Box \alpha$
C1	$\vdash$	$\Box \forall x \alpha \to \Box \alpha$	C5	$\vdash$	$\forall x \Box \alpha \to \Box \forall x \Box \alpha$
$\forall$-Gen	$\vdash$	$\Box \forall x \alpha \to \forall x \Box \alpha$	C1, SV	$\vdash$	$\forall x \Box \alpha \to \Box \forall x \alpha$

We will now show how the axiom QX0 allows additional variables to be added to a proof. As an example we prove

(3) $\vdash \exists x(\neg x \wedge \bigcirc x \wedge \bigcirc\bigcirc\Box\neg x)$

QX0, T-Gen, C2	$\vdash$	$\exists x(x \wedge \bigcirc\Box\neg x) \wedge \bigcirc\exists y(y \wedge \bigcirc\Box\neg y)$
QX1, PLTL, QPL	$\vdash$	$\exists x\exists y(\neg(y \wedge \neg x) \wedge \bigcirc(y \wedge \neg x) \wedge \bigcirc\bigcirc\Box\neg(y \wedge \neg x))$
QX2, PLTL, QPL	$\vdash$	$\exists x(\neg x \wedge \bigcirc x \wedge \bigcirc\bigcirc\Box\neg x)$

A similar proof can be found for all tautologies of the form

(4) $\exists x(\pm x \wedge \ldots \wedge \bigcirc^n \pm x \wedge \bigcirc^{n+1}\Box\neg x)$.

We will refer to the proof of all such tautologies as *finite marking* (FM). This class of tautologies can be used to dictate the interpretation of a quantified variable over the finite future. We note that finite marking is not strong enough to prove tautologies of the form $\vdash \exists x\Box(x \leftrightarrow \neg\bigcirc x)$. Such proofs require the new rule QI.

Unfortunately we have no room to prove the following tautology, that states that if there is some assignment of propositions that makes a formula true in the future, there is some assignment of propositions that makes that formula true at the earliest possible moment.

(5) $\vdash \Diamond\alpha(X) \rightarrow \exists X\forall Y((\neg\alpha(Y))\mathcal{U}\alpha(X)))$.

4 Proof of completeness

In this section we will prove that **QF** is sound for QPTL and we will state that it is complete for QPTL. The full details of the completeness proof stretch over most of the rest of the paper. However, here we are able to easily reduce completeness to a question about the expressiveness of the fragment which consists just of PLTL formulas under existential quantification. In fact, we are left only to show that each QPTL formula has some existentially quantified equivalent and that our proof system can establish this equivalence.

LEMMA 1 *The proof system* **QF** *is sound for QPTL.*

Proof. Since the semantics of QPTL directly extend the semantics of PLTL, the soundness of the axioms C0-C5 and the rules MP and T-Gen can be assumed as they are known to be sound for PLTL. The proof is given by induction over the complexity of proofs so we only have to show that this induction holds for the remaining two axioms and two rules. As this is trivial for both axioms and $\forall$-Gen, we need only show that QI is sound.

Suppose β has temporal depth 1 in X and

(6) $\beta \rightarrow \exists Y(\bigcirc^{\leq 1}(Y \leftrightarrow X) \wedge \bigcirc\beta[Y\backslash X])$

is valid. Therefore if $\sigma \models \beta$ then σ has an X-variant σ' such that $\sigma_0 = \sigma'_0$, $\sigma_1 = \sigma'_1$ and $(\sigma', 1) \models \beta(X)$. Since (6) is a validity we can repeat the argument to get an X-variant of $(\sigma', 1)$, say $(\sigma'', 1)$. We can prefix $(\sigma'', 1)$ with any set of variables, so by repeating this process we can construct the sequence $\sigma^* = (\sigma_0, \sigma'_1, \sigma''_2, \ldots)$. since β has temporal depth 1 in X its interpretation at $(\sigma^*, 0)$ depends only on σ_0 and σ'_1. By construction $(\sigma^*, i) \models \beta$ for all i and σ^* is an X-variant of σ. Therefore $\sigma \models \exists Y \Box \beta[Y \backslash X]$ and we are done. ■

We define a Σ_1 formula (or an existentially quantified formula) to be of the form $\exists X \alpha$ where α is a PLTL formula, and X is a finite set of propositions

LEMMA 2 (Main Lemma) *Every QPTL formula α is provably equivalent in* **QF** *to an existentially quantified formula.*

This will be proven in Sections 6 and 7 and most of what follows leads to this result.

LEMMA 3 *A PLTL formula α is satisfiable if and only if $\exists X \alpha$ is satisfiable.*

Proof. If α has a PLTL model σ then σ is clearly also a model of $\exists X \alpha$. Conversely if $\exists X \alpha$ has some model σ then there is an X-variant of σ that is a model of α. ■

THEOREM 4 **QF** *is complete for QPTL.*

Proof. We know that the satisfiable formulas of PLTL are exactly the consistent ones [4], and hence an existentially quantified formula of QPTL is satisfiable if and only if it is consistent (Lemma 3). Since every formula α of QPTL is provably equivalent to an existentially quantified formula in **QF** (Lemma 2), the satisfiability of α is equivalent to the consistency of α and thus **QF** is complete. ■

5 ω-automata

Our completeness proof, like that in [5], will make serious use of automata. We will use an induction on the complexity of formulas to show the equivalence of QPTL formulas with existentially quantified ones. The case of negation is difficult and we resort to complementation of automata to carry this through. In this section we provide the background definitions and some results relating automata to QPTL and our proof system.

It has been shown [15] that the language QPTLP is equivalent to the language of ω-automaton. We will use this equivalence to show that **QF** is complete. More specifically, we will represent deterministic Rabin (DR) and non-deterministic Büchi (NB) automaton in the language by way of a *characteristic formula*. In our proof we are not only required to know every formula of QPTL is equivalent to an ω-automaton, we also need to show that such an equivalence is a theorem of **QF**.

We will first establish some notation. An automaton is given by the tuple $(\Sigma, Q, q_0, \rho, \mathcal{C})$. The automaton acts on the alphabet, $\Sigma = \wp(\mathcal{L})$ for some finite $\mathcal{L} \subset \mathcal{V}$. The automaton will consist of a finite set of states, Q, and an initial state $q_0 \in Q$ is specified. The transition function, ρ is given by $\rho = \delta : Q \times \wp(\mathcal{L}) \longrightarrow Q$ (a deterministic transition function) or $\rho = \lambda : Q \times \wp(\mathcal{L}) \longrightarrow \wp(Q)\backslash\emptyset$ (a non-deterministic transition function).

We consider a run, $r = (r_0, r_1, \ldots)$, of an automaton over a structure σ to be a sequence of states taken from Q where $r_0 = q_0$ and $r_{i+1} = \delta(r_i, (\sigma_i \cap \mathcal{L}))$, (or $r_{i+1} \in \lambda(r_i, (\sigma_i \cap \mathcal{L}))$, in the case of nondeterministic automaton). We denote the set of states that occur infinitely often in r as $inf(r)$.

In the case of Rabin automata the acceptance condition, $\mathcal{C}$, is given as a set of pairs $\{(L_i, U_i) | i = 1, \ldots, k\}$ where $L_i, U_i \subseteq Q$. A run r satisfies the acceptance condition if for some i, $inf(r) \cap L_i \neq \emptyset$ and $inf(r) \cap U_i = \emptyset$. The Büchi acceptance condition is given as $\mathcal{C} = F \subseteq Q$ and a run r satisfies the condition if $inf(r) \cap F \neq \emptyset$.

We say a formula α of QPTL is congruent to some ω-automaton A when A accepts some structure σ if and only if $\sigma \models \alpha$. We state without proof that every formula of QPTL is congruent to some DR-automaton and some NB-automaton, [18]. However we will show by construction that every ω-automaton A is congruent to some QPTL formula χ_A, the characteristic formula of A.

Suppose the automaton A is given as above. We let $Q = \{q_0, \ldots, q_{n-1}\} \subset \mathcal{V}\backslash\mathcal{L}$. While the states and atomic variables representing the states have the same names, the intended meaning is always clear from the context. We define the formula at_q_i to be $q_i \wedge \neg \bigvee_{j \neq i} q_j$ and given a set of atoms $R = \{r_0, \ldots, r_{n-1}\}$ we let $at\_q_i(R)$ be an abbreviation for $at\_q_i[R\backslash Q]$. This notation also extends to the following formulas.

For each subset $P \subseteq Q$ we define the formula in_P to be $\bigvee_{q_i \in P} at\_q_i$. Finally for each subset $a \subseteq \mathcal{L}$ we define the formula $\rho(a)$ to be $\bigwedge_{x \in a} x \wedge \bigwedge_{x \in \mathcal{L}\backslash a} \neg x$. We can now put these formulas together to simulate a run of the DR-automaton A. We define

$$run_A \quad : \quad \Box \bigvee_{r \in Q}^{a \subseteq \mathcal{L}} (at\_r \wedge \rho(a) \wedge \bigcirc at\_\delta(r,a)) \tag{7}$$

$$acc_A \quad : \quad \bigvee_{1 \leq i \leq k} (\Box \Diamond in\_L_i \wedge \Diamond \Box \neg in\_U_i) \tag{8}$$

$$\chi_A \quad : \quad \exists Q(at\_q_0 \wedge run_A \wedge acc_A). \tag{9}$$

If A were a NB-automaton it is easy to see that a variation of χ_A would suffice as a characteristic formula. The proof that the characteristic formula are congruent to the automaton A is left to the reader. We can derive the following theorem that is analogous with the rule AA of [13].

LEMMA 5 *For every DR-automaton A we can have* $\vdash \exists Q(at\_q_0 \wedge run_A)$.

6 The Σ_1 normal form

The proof of completeness will have two main steps. We will show that every formula of QPTL is equivalent to an existentially quantified formula of PLTL. We do so by induction on the construction of α showing that $O\exists X\alpha$ is provably equivalent to a Σ_1 formula for any operator O $\vee$, $\exists$, $\Diamond$ and $\neg$ and each α from PLTL. The only complicated case is the operator $\neg$, which we will look at now.

To show that Σ_1 is closed under negation, let $\exists Y\alpha$ be some Σ_1 formula. We must show that $\neg\exists Y\alpha$ is provably equivalent to some Σ_1 formula. To do this we will create a series of automata where

- A is a DR-automaton corresponding to α
- B is a NB-automaton accepting exactly the sequences that A does, hence also corresponding to α.
- C is a NB-automaton accepting every Y-variant of the sequences that B does, hence corresponding to $\exists Y\alpha$.
- D is a NB-automaton accepting exactly the sequences that C does not. Hence corresponding to $\neg\exists Y\alpha$.

We show $\vdash \chi_C \leftrightarrow \exists Y\alpha$ and $\vdash \chi_D \leftrightarrow \neg\exists Y\alpha$. Since χ_D is a Σ_1 formula the proof will follow. Let A be a DR-automaton corresponding to α. The first step is easy:

LEMMA 6 $\vdash \chi_A \leftrightarrow \alpha$.

We now create the NB-automaton B that accepts exactly the sequences that A does. We define the automaton B to have the set of states $Q^0 \cup (Q^1 \backslash U_1^1) \cup \ldots \cup (Q^k \backslash U_k^k)$ where the Q^i etc are disjoint copies of Q. The transition relation is given as $\lambda^B(q_i^0, a) = \{\delta(q_i, a)^j | 0 \leq j \leq k\}$ and $\lambda^B(q_i^\ell, a) = \{\delta(q_i, a)^\ell\}$. The accepting set is $\bigcup_{0<i\leq k} L_i^i$. Essentially the automaton B guesses which accepting pair to use and when to use it. This is straightforward:

LEMMA 7 $\vdash \chi_B \leftrightarrow \alpha$.

We now construct a NB-automaton C that accepts exactly all Y-variants of any sequence that B accepts. The states, initial state and acceptance condition of C are identical to those of B. The only change is in the alphabet it reads and the transition function. The alphabet is $\mathcal{L}' = \mathcal{L} \backslash Y$ and we define

$$\lambda^C(r, a') = \bigcup \{\lambda^B(r, a) | a \cap \mathcal{L}' = a'\}.$$

Essentially a run of C is a run of B that ignores the interpretations of $y_0, \ldots, y_m$ and it is easy to see that the construction is both sound and complete. We state the result for $Y = \{y\}$ and the general result can be found by recursively applying this lemma (proof omitted).

LEMMA 8 $\vdash \chi_C \leftrightarrow \exists y \alpha$.

To derive the automaton D we require a complementation procedure for Büchi automata. This is the most complicated part of the proof and has been left to subsection 7.2. We state the lemma here.

LEMMA 9 $\vdash \neg \exists Y \alpha \leftrightarrow \chi_D$

Assuming the proof of Lemma 9 we can now give the proof of the main lemma, (that every formula of QPTL is provably equivalent to a Σ_1 formula).

Proof. (the proof of Lemma 2)

We have mentioned already that we can establish the substitution rule for **QF** using QX2 (amongst other rules and axioms). This is particularly useful for this lemma.

The basic idea is to work within the fragment generated by $\vee$, $\neg$, $\bigcirc$, $\Diamond$ and $\exists x$. It is clear that every formula has an equivalent in this fragment and the equivalence can be shown in **QF**.

We work within this fragment because the result can then be shown by induction on construction of the (rewritten) formula and all but one case is straightforward.

The cases of atoms and formulas $\exists x\alpha$ are trivial. The case of $\vee$ follows as $\vdash ((\exists x\alpha) \vee \beta) \leftrightarrow \exists x(\alpha \vee \beta)$ which itself is established by substitution from semantic validification. The case of $\bigcirc$ can be shown via C3 and QX1. The case of $\diamondsuit$ follows from the proof of (2).

The only difficult case is that for negation and that is exactly what we have provided in Lemma 9: χ_D is a Σ_1 formula.

■

7 Complementation

The complementation problem for non-deterministic Büchi automata has been studied extensively, and several different constructions have been given [15, 14, 7]. To prove the Lemma 9 we give a complementation construction for Büchi automata and prove the correctness of the construction in the proof system **QF**. This construction is effectively a refinement of the construction given by Nils Klarlund [7] and has a slightly better complexity, though both are optimal up to a constant factor in the exponent [11]. The proof of the correctness is significantly different to the one given by Klarlund, and easier to express in terms of proof system, **QF**.

The construction will be given in two steps. First we define a game for which the existence of a winning strategy for one player is equivalent to the automata being unable to accept some model. We then can derive a NB-automata that accepts a model if and only if the said player has a winning strategy. Thus it is the complement of the original automata.

7.1 The game

Given the NB-automaton $A = (\Sigma, Q, q_0, \lambda, F)$ we define the game G by the tuple $(A, N, \mathbb{C}, \Gamma)$ where

- A is the automaton we are seeking to complement.
- N is the linearly ordered set of labels $\{0, \ldots, n\}$ where $|Q| = n$.
- $\mathbb{C}$ is the set of colours $\{\mathbb{R}, \mathbb{G}\}$ (respectively *red* and *green*).
- $\Gamma : Q \times \mathbb{C} \times N^2 \longrightarrow \mathbb{C}$ is the colouring function, where
  - $\Gamma(q, c, i, j) = \mathbb{G}$ if $j > i$,
  - $\Gamma(q, c, i, j) = \mathbb{R}$ if $j \leq i$ and $q \in F$,
  - $\Gamma(q, c, i, j) = c$ otherwise.

The game is played by two players: $\mathcal{A}$, the *automaton* who is trying find a sequence of the states Q that satisfies the accepting condition of A; and

$\mathcal{B}$, the *complement* who is trying to show that A will not be able to satisfy the acceptance condition. The game is played over a *board* $\sigma \in \Sigma^\omega$.

We let $S = Q \times N \times \mathbb{C}$ be the set of *game states*, and define a play of the game G over the board σ to be an infinite sequence $s \in S^\omega$ where:

1. $s_0 = (q_0, 0, \mathbb{R})$
2. If $s_i = (q_i, j_i, c_i)$ and $s_{i+1} = (q_{i+1}, j_{i+1}, c_{i+1})$, then
   (a) $q_{i+1} \in \lambda(q_i, \sigma_i)$,
   (b) $j_{i+1} \in \{k \geq j_i | k \in N\}$,
   (c) $c_{i+1} = \Gamma(q_i, c_i, j_i, j_{i+1})$.

Throughout a play of the game we assume player $\mathcal{A}$ chooses q_{i+1} from the set $\lambda(q_i, \sigma_i)$ and player $\mathcal{B}$ chooses j_{i+1} from the set $\{k \geq j_i | K \in N\}$. Player $\mathcal{B}$ wins if for infinitely many i, $c_i = \mathbb{G}$, and player $\mathcal{A}$ wins otherwise.

We now define the notion of a winning strategy. Given a play s of a game G over σ we write $s_i = (p_i, j_i, c_i)$.

DEFINITION 10 A *strategy* for player $\mathcal{A}$ in the game G over σ is a map $\delta : S \times \mathbb{N} \to Q$ where $\delta(s_i, i) \in \lambda(p_i, \sigma_i)$ for every s_i belonging to some play s such that for all $k \leq i$, $p_k = \delta(s_{k-1}, k-1)$.

A *strategy*, ξ, for player $\mathcal{B}$ is a map $\xi : S \times \mathbb{N} \to \{0, \dots, n\}$ where $\xi(s_i, i) \geq j_i$,for every s_i belonging to some play s such that for all $k \leq i$, $j_k = \xi(s_{k-1}, k-1)$.

A play s is played according the strategies δ and ξ, if for all i, $s_{i+1} = (\delta(s_i, i), \xi(s_i, i), e)$ where e is the colour determined by the game G. A *winning strategy* for a player is a strategy that will ensure the player wins, regardless of the moves of the opponent.

We note that a strategy can depend on σ_k for all k. It is clear that for any σ, $\mathcal{A}$ and $\mathcal{B}$ cannot both have a winning strategy, and it is a consequence of the proofs below that either $\mathcal{A}$ or $\mathcal{B}$ has a winning strategy.

We now show that player $\mathcal{A}$ has a winning strategy in the game G over σ if and only if A accepts σ, and A accepts σ if and only if player $\mathcal{B}$ does not have a winning strategy. To this we require the following definitions and lemmas.

Given the run r of A over σ let δ^r be the strategy defined by $\delta(s_i, i) = r_{i+1}$. It is easy to verify that this is a valid strategy.

LEMMA 11 *If r is an accepting run for A over σ then the strategy δ^r is a winning strategy for $\mathcal{A}$ in the game G over σ.*

DEFINITION 12 A run r of A over σ is *non-accepting at* i if there does not exist some run t with some $k > j > i$ such that $t_i = r_i$, $t_j \in F$ and $t_k = r_k$.

LEMMA 13 *A does not accept σ if and only if for all runs r of A over σ, for all j there is some $i > j$ such that r is non-accepting at i.*

Proof. ($\longleftarrow$) Suppose A accepts σ. Then there is an accepting run r of A over σ, and hence $\forall i$, $\exists k > i$ such that $r_k \in F$. Therefore we can let the run $t = r$ and then for all i we have $t_i = r_i$ and $t_j = r_j \in F$ for some $j > i$. Thus r is not non-accepting for any i and the proof follows by the contrapositive.

($\longrightarrow$) Suppose there is some run r of A over σ and some j such that for all $i > j$, r is not non-accepting at i. Then for all $i > j$ there exists some run t^i and some $j^i > i$ such that $t^i_i = r_i$, $t^i_{j^i} = r_{j^i}$ and $t^i_k \in F$ for some k where $i < k < j^i$.

We define the sequence $I = (i_0, i_1, \ldots)$ inductively by $i_0 = j$, and $i_{m+1} = j^{i_m}$. We can now define an accepting run u, of A over σ that is accepting with respect to ϕ as:

$$
\begin{array}{ll}
u_k = r_k & \text{for } k < i_0 \\
u_k = t^{i_n}_k & \text{for } i_n \le k < j^{i_n}, \text{for all } n \\
u_k = r_k & \text{for } j^{i_n} \le k < i_n, \text{ for all } n
\end{array}
$$

Since r and t^{i_n} are valid runs, and $t^i_i = r_i$, $t^i_{j^i} = r_{j^i}$, it follows that u is a valid run. Furthermore, from the construction of u it follows that for all m, there is some k such that $i_m < k < i_{m+1}$ and $u_k \in F$, and thus u is an accepting run. Therefore A accepts σ and we are done. ∎

We can now define the strategy ξ for player $\mathcal{B}$ as

1. $\xi(s, 0) = 0$ for all $s \in S$.

2. if $i > 0$ and $s = (p, j, c)$ then

$$
\begin{array}{ll}
\xi(s, i) = j + 1 & \text{if } c = \mathbb{R},\ j < n, \text{ and } p = r_i \\
 & \text{for some run } r, \text{ non-accepting at } i, \\
\xi(s, i) = j & \text{otherwise.}
\end{array}
$$

LEMMA 14 *Given the strategy ξ, for any run r of A over σ if for all j, there exists $i > j$ such that r is non-accepting at i, then ξ will win against the strategy δ^r in the game G.*

Proof. Let $s = (p_i, j_i, c_i)_{i \in \mathbb{N}}$ be a play of G according to the strategies ξ and δ^r, where r is non-accepting infinitely often. Suppose for some i, $c_i = \mathbb{R}$. Then there must be some least $k > i$ such that r_k is non-accepting at k. According to the strategy ξ, $c_k = \mathbb{R}$ and provided $j_k < n$, $c_{k+1} = \mathbb{G}$.

Thus, to show that player $\mathcal{B}$ wins we must simply show that for all i, $j_i < n$. Suppose, for contradiction, that $j_i = n$. Then there must be n indices $k_1 < k_2 < \ldots < k_n$ where for $a \leq n$, $r_{k_a} = r^a_{k_a}$ for some run r^a that is non-accepting at k_a and $c_{k_a} = \mathbb{R}$. For all $a < n$ we have $c_{k_a+1} = \mathbb{G}$ and $c_{k_{a+1}} = \mathbb{R}$, and thus there is some m where $k_a < m < k_{a+1}$ and $r_m \in F$. Since r^a is non-accepting at k_a then for all $\ell > m$, for all runs t with $t_m = r_m$ we have $r^a_\ell \neq t_\ell$, and $r^a_\ell \notin F$.

Thus if $j_i = n$ then there are $n-1$ runs $r^1, \ldots, r^{n-1}$, such that for all $i > k_{n-1}$, $r^a_i = r^b_i$ if and only if $a = b$. However since $|Q| = n$, $|F| > 0$ and $r^a_i \notin F$ it must be that $|F| = 1$ and for all $\ell > k_{n-1}$ we must have $r_\ell \in F$, contradicting the fact that r is non-accepting infinitely often. ■

7.2 The automaton

Given the game $G = (A, N, \mathbb{C}, \Gamma)$, we now construct an automaton $\overline{A}$, where a run of $\overline{A}$ over σ corresponds to a set of plays of G over σ. We will show that there is an accepting run of $\overline{A}$ over σ if and only if player $\mathcal{B}$ has a winning strategy for G over σ, and hence $\overline{A}$ is a complement of A.

The states of the automaton $\overline{A}$ are maps from the set of states Q (of the automaton A) to a set of labels, $\{0, \ldots, n\} \times \mathbb{C}$ where $\mathbb{C}$ is the set of colours $\mathbb{C} = \{\mathbb{R}, \mathbb{A}, \mathbb{G}\}$ (*red, amber* and *green*). We define an ordering, $<$, over the labels where $(j, c) < (j', c')$ if

1. $j < j'$ or,
2. $j = j'$ and $c = \mathbb{G} \neq c'$ or
3. $j = j'$ and $c = \mathbb{A}$ and $c' = \mathbb{R}$.

We define the automata $\overline{A} = (\overline{Q}, \phi_0, \overline{\lambda}, \overline{F})$ where

- $\overline{Q} = \{\phi | \phi : Q \to [0, n] \times \mathbb{C}\}$ where $|Q| = n$.
- $\phi_0(q_0) = (0, \mathbb{R})$ and $\phi_0(q) = (0, \mathbb{G})$ for $q \neq q_0$.
- $\overline{F} = \{\phi | \phi(Q) \subseteq [0, n] \times \{\mathbb{A}, \mathbb{G}\}\}$

and for each $\phi \in \overline{Q}$ and $a \in \mathcal{L}$ we define the set $\overline{\lambda}(\phi, a) \subset \overline{Q}$ by $\psi \in \overline{\lambda}(\phi, a)$ if and only if for all $p, q \in Q$ with $q \in \lambda(p, a)$:

1. $\phi(p) \leq \psi(q)$;

2. if $\phi(p) \in [1, n] \times \{\mathbb{G}\}$ and $p \in F$ then $\phi(p) < \psi(q)$;

3. if $\phi(p) \in [1, n] \times \{\mathbb{A}\}$ and $\phi \in \overline{F}$ then $\phi(p) < \psi(q)$.

LEMMA 15 *If θ is an accepting run of $\overline{A}$ over σ then player $\mathcal{B}$ has a winning strategy over σ.*

Proof. We give this proof by constructing the winning strategy δ^* from the run θ. Let $s_i = (p_i, j_i, c_i)$. Then

$$\delta^*(s_i, i) = k, \text{ where } \theta_i(p_i) = (k, c^*) \text{ for some } c^* \in \mathbb{C}.$$

To show that δ^* is a valid strategy we have to show that $\delta^*(s_i, i) \geq j_i$ for all $i > 0$ in every play. By the definition of δ^* we have $\theta_{i-1}(p_{i-1}) = (j_i, c')$, for some c', and we also know that $p_i \in \lambda(p_{i-1}, \sigma_{i-1})$. It follows from the definition of $\overline{\lambda}$ that $\theta_i(p_i) \geq \theta_{i-1}(p_{i-1})$ and hence $\delta^*(s_i, i) \geq j_i$.

Now suppose for contradiction that δ^* were not a winning strategy. Player A could chose some run r that wins against the strategy δ^*. If s were the resulting play there would be some fixed j and some least k such that for all $i \geq k$, $s_i = (r_i, j, \mathbb{R})$. If $j = 0$, then it is clear that $\theta_i(r_i) = (0, \mathbb{R})$ for all $i \geq k$. If $j > 0$ then $s_k = (r_i, j, \mathbb{R})$ and, since k is least, $s_{k-1} = (r_{k-1}, j, \mathbb{G})$. Therefore $\theta_{k-1}(r_{k-1}) \geq (j, \mathbb{G})$ and $r_{k-1} \in F$, so by the definition of $\overline{\lambda}$ it follows that for all $i \geq k$, $\theta_i(r_i) \geq (j, \mathbb{A})$.

Using the definition of the strategy δ^*, we show that for all $i \geq k$, $\theta_i(r_i) \in \{j\} \times \{\mathbb{A}, \mathbb{R}\}$. To see that $\theta_i(r_i)$ cannot be green for any $i \geq k$ we must consider two possibilities. If $j = 0$ then $\theta_i(r_i) = (0, \mathbb{R})$ as r_i is the state of some run at i and thus greater than $\theta_0(q_0)$. If $j > 0$ then $s_k = (r_i, j, \mathbb{R})$ and, since k is least, $s_{k-1} = (r_{k-1}, j, \mathbb{G})$. Therefore $\theta_{k-1}(r_{k-1}) \geq (j, \mathbb{G})$ and $r_{k-1} \in F$. From the definition of $\overline{\lambda}$ it follows that for all $i \geq k$, $\theta_i(r_i) \geq (j, \mathbb{A})$.

If for any $m > k$, $\theta_m \in \overline{F}$ then for all $i > m$, $\theta_i(r_i) = (j, \mathbb{R})$ and thus $\theta_i \notin \overline{F}$. Therefore θ is not an accepting run giving us the desired contradiction. ∎

LEMMA 16 *If player $\mathcal{B}$ has a winning strategy over σ then there is an accepting run of $\overline{A}$ over σ.*

Proof. Given δ^*, a winning strategy for player $\mathcal{B}$, we construct the run θ in several steps. First we define a sequence $\eta = (\eta_0, \eta_1, \ldots)$ where for all i, $\eta_i : Q \longrightarrow \{0, \ldots, n\}$. For each $p \in Q$ and each i we define

$$\eta_i(p) = \max\{0, j | s_i = (p, j, c) \text{ for some } c \text{ and some play } s \text{ according to } \delta^*\}.$$

We then define a sequence $\gamma = (\gamma_0, \gamma_1, \ldots)$ where for all i, $\gamma_i : Q \longrightarrow \mathbb{C}$. To do this for all $p \in Q$ and for all i, we let $\mu(p,i) \subseteq Q$ be the set of *minimal successors* of p at i, defined as $\{q | q \in \lambda(p, \sigma_i) \text{ and } \eta_i(q) = \eta_i(p)\}$. We now define the sequence γ with the following induction:

- $\gamma_0(q_0) = \mathbb{R}$ and $\gamma_0(q) = \mathbb{G}$ for all $q \neq q_0$.
- for every $p \in Q$
    1. if $p \in \mu(q,i)$ for some q such that either
        (a) $\gamma_i(q) = \mathbb{R}$, or
        (b) $\gamma_i(q) = \mathbb{A}$ and $\gamma_i(Q) \subseteq \{\mathbb{G}, \mathbb{A}\}$
        then $\gamma_{i+1}(q) = \mathbb{R}$,
    2. otherwise if $p \in \mu(q,i)$ for some q such that either
        (a) $\gamma_i(q) = \mathbb{A}$, or
        (b) $\gamma_i(q) = \mathbb{G}$ and $q \in F$
        then $\gamma_{i+1}(q) = \mathbb{A}$,
    3. otherwise $\gamma_{i+1}(q) = \mathbb{G}$.

We define a run θ, of $\overline{A}$ over σ by $\theta_i(p) = (\eta_i(p), \gamma_i(p))$ for all $p \in Q$ and all i. To see that θ is a valid run of $\overline{A}$ over σ we observe that if $q \in \lambda(p, \sigma_i)$ then $\eta_{i+1}(q) \geq \eta_i(p)$ and the rest follows from the inductive definition of γ.

Now suppose, for contradiction, that θ is not an accepting run. Therefore there are only finitely many accepting states, so we let m be the index one greater than the last accepting state (i.e. for all $i \geq m$, $\theta_i \notin \overline{F}$).

By the inductive definition of γ we can see that if $\gamma_i(p) = \mathbb{R}$ for some $i \geq m$ then there is some q with $p \in \lambda(q, \sigma_{i-1})$ with $\gamma_{i-1}(q) = \mathbb{R}$. Applying this recursively we can deduce that there is some run r of A over σ with $r_i = p$ and for all k with $m \leq k \leq i$, $\gamma_k(r_k) = \mathbb{R}$, and hence $\eta_k(r_k) = \eta_i(r_i)$.

For all $k \geq m$ and for all $i \geq k$ let

$$P^j(k,i) = \{r_k \in Q | r \text{ is a run of } A \text{ with } \theta_k(r_k) = \theta_i(r_i) = (j, \mathbb{R})\}.$$

The set $P^j(k,i)$ is the set of states that belong to some run at i that has had the label $(j, \mathbb{R})$ since k. We choose $j \leq n$ to be such that for all $k \geq m$ and for all $i > k$, $P^j(k,i) \neq \emptyset$. Such a j exists since $P^j(k,i) = \emptyset$ implies either $P^j(k, i+1) = P^j(k+1, i) = \emptyset$ or $\theta_k \in \overline{F}$. Since $\theta_k \notin \overline{F}$ for $k > m$, if such a j did not exist there would be some value $\ell > m$ such that $P^j(\ell, \ell) = \emptyset$ for all j. Consequently we would have $\theta_\ell \in \overline{F}$, contradicting the fact that $\theta_k \notin \overline{F}$ for all $k > m$.

We can define a losing play of the game for $\overline{A}$ as follows. Since $|Q|$ is finite there must be some $q_m \in Q$ such that $q_m \in P^j(m, i)$ for infinitely many i. By the definition of θ we can assume that there is some play s, played according to δ^*, with $s_m = (q_m, j, \mathbb{R})$. Since $q_m \in P^j(m, i)$ for infinitely many i, there must be some $q \in \lambda(q_m, \sigma_m)$ such that $q \in P^j(m+1, i)$ for infinitely many i. Player A can choose $q_{m+1} = q$ and by the way that θ was constructed we can see that $\delta^*(s_m, i) = j$, so $s_{m+1} = (q_{m+1}, j, \mathbb{R})$. Since $q_{m+1} \in P^j(m+1, i)$ for infinitely many i, the same argument applies and player A can ensure that $s_{m+2} = (q_{m+2}, j, \mathbb{R})$. Continuing this way player A can ensure that for all $i > m$ $j_i = j$ and $c_i = \mathbb{R}$, and hence win the game, contradicting the fact that δ^* is a winning strategy. Therefore θ must be an accepting run and we are done. ■

COROLLARY 17 *The automaton $\overline{A}$ accepts σ if and only if the automaton A does not.*

COROLLARY 18 *Given a non-deterministic Büchi automaton with n states, we can find a complementary non-deterministic Büchi automaton with $(3(n+1))^n$ states.*

This follows directly from the complementation construction. Furthermore, a close examination of the proof of correctness will show that the complementary automaton need only use the states

$$\{\phi | \phi : Q \longrightarrow \{1, \ldots, n-1\} \times \mathbb{C} \cup \{(0, \mathbb{G}), (0, \mathbb{R}), (n, \mathbb{R})\}\}.$$

We can therefore improve the complexity of this construction to $(3n)^n$. This is a slight improvement on Klarlund's construction [7] which has complexity $(4n)^n$.

7.3 Formal proof of correctness

Because of space limitations we omit the part of the proof which shows that the correctness of the complementation construction can be established in our proof system. Details will appear in the full paper.

8 Redundancy

We consider whether any parts of the axiom system are redundant. The two candidates are the new axiom QX0 and the rule QI.

To show that QX0 is necessary we consider an alternative "wrapped" semantics for QPTL formulas based on Kripke structures. Even consideration of runs on a one state Kripke structure will suffice. Suppose $(S, R) = (\{s_0\}, \{(s_0, s_0)\})$ a structure with one state $s_0 \subseteq \mathcal{V}$ and a transition relation R which allows a transition from s_0 to s_0.

Now consider sequences σ of states from S, i.e. $\sigma = (s_0, s_0, s_0, ...)$. We can redefine the semantics for QPTL, i.e. give a "wrapped" semantics, by using the same clauses for each operator as in the standard semantics apart from the case of $\forall y\alpha$. Instead define $(\sigma, i) \models \forall y\alpha$ iff for every structure (S', R) which differs from (S, R) only to the extent that $s_0 \setminus \{x\} = s'_0 \setminus \{x\}$, we have $(\sigma, i) \models \alpha$.

It is straightforward to show that QF without the QX0 rule is sound for these wrapped semantics: so anything derivable in this system is a validity of wrapped QPTL. However, $\exists x(x \wedge \Box\neg x)$ is clearly not sound itself and so not derivable in the rest of the system.

In order to show that QI is not redundant we again can change the semantics of the logic. Briefly, we only consider models in which atoms change truth value a finite number of times as we proceed through time. It can be shown that **QF** is sound for such a language, and the result follows. Full details will appear in the long version of the paper.

9 Past operators

As in [19], the traditional unquantified PLTL is often extended to a logic which we call PLTLP (although there is no standard name) by the addition of past-time temporal operators. The extension brings no extra expressivity (in a certain sense [4]) but there are technical reasons and advantages in practical convenience. We can also add these operators to quantified propositional temporal logic to form a logic QPTLP. The syntactic formation rules for QPTLP are simply those for QPTLalong with allowing $\circleddash\alpha$ and $\boxminus\alpha$ if α and β are formulas.

The new operators are **weak previous** $\circleddash$ and **past generally** $\boxminus$. Add the following semantic clauses:

$$(\sigma, j) \models \circleddash\alpha \Leftrightarrow (j = 0) \text{ or } (\sigma, j-1) \models \alpha. \tag{10}$$

$$(\sigma, j) \models \boxminus\alpha \Leftrightarrow \forall k \leq j \quad (\sigma, k) \models \alpha. \tag{11}$$

Kesten and Pnueli's proof system [6] for QPTLP consists of axioms for PLTL, axioms for the past-time operators, extra PLTLP axioms describing the interaction of the past and future operators, QX1, QX2, MP, T-GEN and $\forall$-GEN. There is nothing like our QX0 or our QI rule. The complicated completeness proof, using automata as we do, is presented in that paper.

It is interesting to consider whether their proof system can be made into a complete system for QPTL by the simple removal of axioms which mention the past operators. The "wrapped" semantics which we introduced above shows that this is not possible: QX0 is not derivable. To derive QX0 one makes use of the Kesten and Pnueli axiom $\circleddash\bot$. Furthermore, an examination of the completeness proof shows heavy reliance on the past-time operators for establishing various marking and inductive constructions. Our desire to avoid such essential and subtle use of the past-time operators led us to instead use the Σ_1 normal form and deterministic Rabin automata approach to our completeness proof.

We have our reasons for not wanting to use the past-time operators. It is worth considering whether there are any good reasons to add them to QPTL as they are not needed to obtain a complete axiom system. The answer, even more clearly than in the PLTL case, is that they are not useful. Not only do they add no extra expressiveness but they are easy to define in terms of the future operators and quantification. We will now be more precise with our claim here.

To proceed we define a translation from QPTLP into QPTL which preserves truth at the start of time but not necessarily at other times. Consider a QPTLP formula α. Recursively replace past operators with new atoms from the bottom up. For each sub-formula $\ominus\beta$ choose a new atom q_β say, replace $\ominus\beta$ by q_β in the formula and add a new conjunct $q_\beta \wedge \Box(\beta \rightarrow \bigcirc q_\beta))$. Similarly choose a new atom p_β to replace each $\boxminus\beta$ and add a new conjunct $(\beta \leftrightarrow p_\beta) \wedge \Box(p_\beta \rightarrow \bigcirc(\beta \rightarrow p_\beta))$. Finally existentially quantify over all the new atoms in the resulting past-free formula. Call the QPTL result α^+ say. A straightforward induction shows the following:

LEMMA 19 *For each $\alpha \in$ QPTLP, for each structure σ, $(\sigma, 0) \models \alpha$ iff $(\sigma, 0) \models \alpha^+$.*

Thus it is easy to find an (initially) equivalent QPTL formula to any QPTLP formula. The translation is PTIME in contrast to the non-elementary process in the PLTL case. Initial equivalence is exactly what is needed for reasoning about specifications of systems.

In the long version of this paper, we also show that we obtain a sound and complete axiomatization for QPTLP by the simple addition to QF of the standard PLTLP axioms for the past time operators.

10 Conclusion

Deciding QPTL like QPTLP is non-elementarily complex [18]. However, it is worth noting, in the context of our reduction to Σ_1 formulas, that the satisfiability problem for existentially quantified QPTL is, like that for PLTL in PSPACE.

The main achievements here are: a sound and complete proof system for pure future quantified propositional temporal logic; a straightforward process for converting QPTL formulas to a Σ_1 normal form; and a new and interesting proof of correctness for an optimal complementation procedure for non-deterministic Büchi automata.

In future work we hope that alternative semantics for QPTL, such as the wrapped semantics, are considered and axiomatized and branching time extensions of QPTL are also axiomatized.

The extension of QPTL to the branching time case is complicated with a variety of possible semantics. See [8] and [2]. However, it is worth pointing out that our new proof system for QPTL allows us to side-step the complications caused by past operators in the move to branching time logic. The use of quantification in a branching time setting is complicated enough. The addition of the past time operators just adds even more subtle variations without any reward.

BIBLIOGRAPHY

[1] K. Engelhardt, R. van der Meyden, and Y. Moses. Knowledge and the logic of local propositions. In *Conf. on Theoret. Aspects of Rationality and Knowledge*. 1998.

[2] T. French. Decidability of quantified propositional branching time logics. In *Proc. of the 14th Aust. Joint Conf. on A. I. (AI'2001)*, pages 165–176, 2001.

[3] T. French. Quantified propositional branching time logics. Thesis, in prep.

[4] D. M. Gabbay, A. Pnueli, S. Shelah, and J. Stavi. On the temporal analysis of fairness. In *7th ACM Symp. on Princ. of Prog. Languages, Las Vegas*, pages 163–173, 1980.

[5] Y. Kesten and A. Pnueli. A complete proof systems for QPTL. In *Proc., 10th Annual IEEE Symp. on Logic in Computer Science*, pages 2–12, 1995.

[6] Y. Kesten, and A. Pnueli. Complete proof system for QPTL. *J. Logic and Computation.* To appear in 2003.

[7] N. Klarlund. Progress measures for complementation of ω-automata with applications to temporal logic. In *Proc. Foundations of Computer Science*, 1991.

[8] O. Kupferman. Augmenting branching temporal logics with existential quantification over atomic propositions. In *Computer Aided Verification, Proc. 7th Int. Conference*, volume 939 of *LNCS*, pages 325–338, Liege, July 1995. Springer-Verlag.

[9] L. Lamport. Specifying concurrent program modules. *ACM Transactions on Prog. Languages and Systems*, 5:190–222, 1983.

[10] L. Lamport. The temporal logic of actions. Tech. Report 79, DEC Systems Research Center, Palo Alto, Calif., USA, 1991.

[11] M. Michel. Complementation is more difficult with automata on infinite words. Manuscript, CNET, Paris, 1988.

[12] A. Pnueli. The temporal logic of programs. In *Proc. of the Eighteenth Symp. on Foundations of Computer Science*, pages 46–57, 1977. Providence, RI.

[13] M. Reynolds. An axiomatization of full computation tree logic. *J. Symbolic Logic*, 63(3):1011–1057, 2001.

[14] S. Safra. On complexity of ω-automata. In *Proc. Found. of Computer Science.* 1988

[15] A. Sistla, M. Vardi, and P. Wolper. The complementation problem for Buchi automata with applications to temporal logic. *Theo. Computer Science*, 49:217–237, 1987.

[16] A. P. Sistla. *Theoretical Issues in the Design and Verification of Distributed Systems.* PhD thesis, Harvard, 1983.

[17] P. Wolper. Temporal logic can be more expressive. *Inf.and Comp.*, 56(1–2):72–99, 1983.

[18] P. Wolper, M. Vardi, and A. Sistla. Reasoning about infinite computation paths. In *Proc. of 24th IEEE Symp. on the Found. of Computer Science*, 1983.

[19] L. Zuck, O. Lichtenstein, A. Pnueli. The glory of the past. *LNCS*, 193:196–218, 1985.

Tim French
Department of Computer Science and Software Engineering
University of Western Australia
35 Stirling Highway
Crawley, Western Australia, 6009
Australia
E-mail: tim@csse.uwa.edu.au

Mark Reynolds
School of Information Technology
Murdoch University
Murdoch, Perth, Western Australia 6150
Australia
E-mail: m.reynolds@murdoch.edu.au