Complete Proof System for QPTL*

Yonit Kesten† Amir Pnueli‡

February 20, 2001

Abstract. The paper presents an axiomatic system for *quantified propositional temporal logic* (QPTL), which is propositional temporal logic equipped with quantification over propositions (boolean variables). The advantages of this extended temporal logic is that its expressive power is strictly higher than that of the un-quantified version (PTL) and is equal to that of S1S, as well as that of ω-automata. Another important application of QPTL is its use for formulating and verifying refinement relations between reactive systems. In fact, the completeness proof is based on the reduction of a QPTL formula into a Büchi automaton, and performing equivalence transformations on this automata, formally justifying these transformations as a bi-directional refinement.

1 Introduction

For a long time, temporal logics have been mainly used for the specification and verification of *properties* of reactive systems. According to this approach, a system is specified by a list of properties, all of which should be satisfied by any acceptable implementation. As long as this was the main use, quantifiers did not play an important role in temporal logic (TL). There were only two places where the use of quantifiers was recommended:

1. Using *quantification over rigid variables* (variables that stay the same throughout the model) to connect values of system variables at two different states. For example, the formula

$$\forall u: \quad x = u \quad \Rightarrow \quad \Diamond(y = u^2)$$

uses quantification over the rigid variable u to specify the property that every value which appears in variable x at some state, appears squared in variable y some later time. This may be used to specify the behavior of a procedure with input x and output y, whose task is to square numbers.

However, very soon one realizes that this formula is valid over a set of program computations iff the same formula without the universal quantification is valid. Therefore one may specify the same property with the simpler formula:

*This research was supported in part by the Minerva Center for Verification of Reactive Systems and by a grant from the U.S.-Israel Binational Science Foundatio

†Contact author: Dept. of Communication Systems Engineering, Ben Gurion University, Israel, ykesten@bgumail.bgu.ac.il

‡Dept. of Applied Mathematics and Computer Science, the Weizmann Institute of Science, Israel, amir@wisdom.weizmann.ac.il

$$x = u \quad \Rightarrow \quad \diamond(y = u^2).$$

2. Using *quantification over flexible variables* (variables that may change from one state to another) in order to increase the expressive power of the logic. For example, it is a known fact (see [Kam68], [GPSS80], and [MP71]) that unquantified (propositional) TL cannot count. In particular, it is impossible to specify in (unquantified) PTL that p must hold on every *even* position in the model. In QPTL, this is specified by the following formula:

$$\exists t:\ t \wedge \square(\bigcirc t \leftrightarrow \neg t) \wedge \square(t \rightarrow p)$$

In this formula, the auxiliary (specification) variable t is used as a counter modulo 2. It is true at all even positions and false at all odd positions.

In spite of these two cases, quantification never played a central role in the use of TL for specification and verification of properties. This changed drastically with the suggestion of using TL for proving *refinement* (or *implementation*) between programs. This suggestion is extensively discussed in the framework of TLA ([Lam94]), which freely interchanges formulas and programs, and has also been studied under the framework of TL considered here [KMP94]. According to this approach, to verify that system S_1 refines system S_2, we have to prove the implication

$$\left(\exists x_1 : sem_{S_1}\right) \quad \rightarrow \quad \left(\exists x_2 : sem_{S_2}\right) \tag{1}$$

where sem_{S_i}, $i = 1, 2$, is a temporal formula called the *temporal semantics* of system S_i ([MP83], [MP91]), which characterizes all models that are computations of S_i, and x_i, $i = 1, 2$, are the *internal variables* of S_i. This formula states that, for every σ_1, a computation of S_1, there exists σ_2, a computation of S_2, such that σ_1 and σ_2 agree on the interpretation of all variables except possibly the internal variables x_1 and x_2. The assumption is that S_1 and S_2 do share some observable variable y on which σ_1 and σ_2 must agree. Implication (1) is valid iff the following implication is valid:

$$sem_{S_1} \quad \rightarrow \quad \exists x_2 : sem_{S_2} \tag{2}$$

The main approach for establishing such an implication is to prove the implication

$$sem_{S_1} \quad \rightarrow \quad sem_{S_2}[x_2 \mapsto t_2], \tag{3}$$

where $sem_{S_2}[x_2 \mapsto t_2]$ is obtained from sem_{S_2} by replacing all occurrences of x_2 by the term t_2 which, in general, may be a function of x_1 and y. We can view $t_2(x_1, y)$ as a *Skolem function* mapping values of x_1 and y into values of x_2. This corresponds to the well-known notion of *refinement mapping*, establishing a mapping between states of S_1 and states of S_2. Methods for proving refinement by refinement mapping have been extensively discussed in the literature, e.g., [Lam83], [LS84], [LT87], [Jon87], and [AL91].

An important question is how to define the Skolem function $x_2 = t_2(x_1, y)$ in an effective way. Observe that t_2 is a temporal function in the sense that the value of t_2 at position j of a model may depend on the values of x_1 and y at positions that either precede or succeed j. In [AL91], Abadi and Lamport identify two definitional schemes for establishing the necessary temporal Skolem function:

- A *history scheme* which can be defined by the forward inductive definition

  $u = f(\ominus u)$

  This scheme defines the value of u at position j as a function of u at position $j-1$.

- A *prophecy scheme* which can be defined by the backward inductive definition

  $u = f(\bigcirc u)$

  This scheme defines the value of u at position j as a function of u at position $j+1$. In [AL91], it is required that f has a finite range. Since we are dealing here with the propositional case, this is always guaranteed.

While the history scheme always yields a unique solution, a prophecy scheme (under the finite range assumption) may have one or more solutions. For example, the scheme $y = \neg \bigcirc y$ has one solution in which $y = 0$ at all even positions and another solution in which $y = 1$ at all even positions. We describe this situation by saying that the prophecy scheme may yield one or more applicable skolem functions. The two schemes correspond to the well-known methods of *forward* and *backward* simulation. The claim that these schemes always define one (for history) or more (for prophecy) temporal function can be formally stated by the following two theorems:

$$\exists u : \Box(u = f(\ominus u)) \quad \text{and} \quad \exists u : \Box(u = f(\bigcirc u)) \tag{4}$$

In view of this important application of quantification, it became apparent that every deductive method for proving program refinement by TL should offer a repertoire of theorems and proof rules for formally dealing with quantifiers. The usual question is how do we recognize that our arsenal of axioms and inference rules is adequate. This is a standard question dealt with in logic under the heading of *completeness*. Unfortunately, it has been shown that, unlike the predicate calculus, full TL with quantifiers cannot be effectively axiomatized. In [Sza86] it has been shown that FTL (first-order TL) admits no finite axiomatization, while [Aba89] showed that it cannot have a recursive axiomatization. Related results were already presented in [ANS79].

Learning that there is no chance that true completeness can be established recursively, there are two possible paths to follow. The first is to search for *relative completeness* in the sense of Cook [Coo78]. A second possibility is to restrict the logic to a simpler fragment, where real completeness is possible. Both directions strive for separation of concerns, trying to untangle the interaction between the temporal operators (or the program dynamics aspects in the case of [Coo78]) and the rich data structures.

In this paper we chose to follow the second path and consider quantified TL where the variables are restricted to range over fixed finite domains. Without loss of generality, we can restrict our attention to boolean variables (propositions). In subsection 4.1 we justify this claim by showing that boolean variables are sufficient to encode a logic over any fixed finite domain. This leads to the version of the logic studied here: *Quantified Propositional Temporal Logic*. We were encouraged to follow this path by the previous successful treatment of the propositional fragment of unquantified TL (PTL) [GPSS80]. Another successful attempt in restricting the logic to achieve real completeness is presented in [HWZ00, WZ00]. The resulting *monodic fragment* of first-order TL, allows variables to range over infinite domain, however any subformula begining with a temporal operator is restricted to a single free variable.

As shown in [SVW87], the satisfiability problem for QPTL is decidable (which is a direct consequence of [Buc62]), albeit with non-elementary complexity [MS73]. Therefore, one may justifiably ask why do we bother to provide an axiomatic system for a decidable logic. Intellectual curiosity aside, our main reason for studying this problem is not because of our interest in QPTL per se but rather that, by studying QPTL, we can gain confidence that the set of axioms and rules we propose are adequate in some restricted cases for reasoning in full QTL, and master techniques for deductive proofs in the presence of quantifiers. The cases for which QPTL reasoning is adequate are those in which we can use abstraction to separate the treatment of data within single states from the dynamics of computations. In these cases, most of the assertions dealing with unbounded data, such as integers and arrays, can be abstracted into propositions.

A source of inspiration for our work came from the monograph of Dirk Siefkes [Sie70], who established completeness of an axiomatic system for S1S after the logic has been shown to be decidable by Büchi. In the introduction, Siefkes explains that he considers Büchi's decision procedure, based on automata, to be semantical (model theoretic) while he was looking for a syntactic (proof theoretic) approach to the same problem.

1.1 An Overview of the Completeness Proof

The full proof of completeness is long and detailed. However, in spite of this technical complexity, the proof is based on a very simple principle. Assume that the temporal formula p is valid. We wish to show that p is provable. Let us apply the algorithm proposed in [SVW87] for checking whether $\neg p$ is satisfiable. The algorithm consists of the following steps applied to an arbitrary formula φ:

1. Construct a Büchi automaton $\mathcal{A}_\varphi$ which accepts precisely the models (infinite sequences) satisfying φ.

2. Check whether $\mathcal{A}_\varphi$ is empty. Formula φ is satisfiable iff $\mathcal{A}_\varphi$ is non-empty.

Applying this algorithm to the formula $\neg p$ which is known to be unsatisfiable (since p is valid), we will obtain an automaton $\mathcal{A}_{\neg p}$ which accepts the empty language. Our completeness proof will mimic the decision algorithm by establishing (denoting provability in our axiomatic system by $\vdash$):

1. $\vdash \varphi \Leftrightarrow \chi_{\mathcal{A}_\varphi}$, where $\chi_{\mathcal{A}_\varphi}$ is the characteristic formula of the automaton $\mathcal{A}_\varphi$, i.e., a formula satisfied by all the sequences accepted by $\mathcal{A}_\varphi$.

2. If $\mathcal{A}_\varphi$ is empty (i.e., accepts the empty language), then $\vdash \neg\chi_{\mathcal{A}_\varphi}$.

Since $\neg p$ is unsatisfiable, $\mathcal{A}_{\neg p}$ will be empty. Consequently, we can prove in our axiomatic system $\vdash \neg p \Leftrightarrow \chi_{\mathcal{A}_{\neg p}}$ followed by $\vdash \neg\chi_{\mathcal{A}_{\neg p}}$, from which we can infer $\vdash p$.

We find it somewhat ironic that, starting with an attempt to establish a proof-theoretic approach to solving the QPTL-validity problem that can serve as an alternative to the automata-theoretic approach initiated by Büchi, the best proof we managed to construct for the completeness of the axiomatic approach is again based on reduction to automata.

Another surprise encountered in the completeness proof is that the history and prophecy theorems (4) which we consider central to a refinement proof need not be taken as axioms but can be proven as theorems. This is one of the first results we obtained by translation of some of the methods developed in [Sie70].

1.2 Comparison with Related Logics

The fact that PTL is strictly less expressive than S1S was established as soon as Kamp showed that PTL is expressively equivalent to the first-order theory of linear order [Kam68]. Many proposals have been made over the years for reasonable enhancements of PTL which will make it as expressive as S1S. Wolper suggested in [Wol83] the logic ETL, introducing grammar operators which add the expressive power of finite-state automata to the logic. Another interesting extension is to add *fix-point* operators to the language. This was considered first in [BB86] by Banieqbal and Barringer, and later by Gabbay [Gab87] where he formulated the logic USF, combining past and future temporal logic with fix-points operators over sequences. The USF logic has been shown to have a complete axiomatization by Hodkinson in [Hod95].

In spite of having these alternative logics which have an expressive power equal to that of QPTL, some of which even possessing complete axiomatic systems, we believe that there is an interest in the study of QPTL and its axiomatization. The main motivation is that the syntax of QPTL has a closer correspondence to the way programming systems are built and reasoned about. In particular, existential quantification corresponds to the introduction of a new local variable which is a very common construct in programming languages, and the addition of auxiliary variables as part of a refinement proof is a long established practice.

2 QPTL: Syntax and Semantics

We assume a countable set of boolean variables (propositions) $\mathcal{V}$. The flow of time considered in this paper is isomorphic with the natural numbers $(0, 1, 2, \ldots)$. The syntax of QPTL formulas is defined as follows:

- Every variable $x \in \mathcal{V}$ is a formula. Each of the constants F and T is a formula.
- If p and q are formulas, then so are

  $$\neg p,\ p \vee q,\ \bigcirc p,\ p \mathcal{U} q,\ \circledcirc p,\ \text{and}\ p \mathcal{S} q$$

- If p is a formula and $x \in \mathcal{V}$ is a variable then

  $$\forall x\ :\ p$$

  is a formula.

Thus, as the set of basic operators we take $\neg$, $\vee$, $\bigcirc$ (next), $\mathcal{U}$ (until), $\circledcirc$ (before, weak previous), $\mathcal{S}$ (since), and $\forall$. Additional operators can be defined by:

$$
\begin{array}{lcllcl}
p \wedge q & = & \neg(\neg p \ \vee \ \neg q) & p \rightarrow q & = & \neg p \vee q \\
p \leftrightarrow q & = & (p \rightarrow q) \ \wedge \ (q \rightarrow p) & & & \\
\diamondsuit p & = & \mathrm{T} \ \mathcal{U} \ p & \diamondsuit\!\!\!-\ p & = & \mathrm{T} \ \mathcal{S} \ p \\
\square p & = & \neg \diamondsuit \neg p & \boxminus p & = & \neg \diamondsuit\!\!\!- \neg p \\
p \mathcal{W} q & = & p \mathcal{U} q \ \vee \ \square p & \ominus p & = & \neg \circledcirc \neg p \\
p \mathcal{B} q & = & p \mathcal{S} q \ \vee \ \boxminus p & & & \\
\widehat{\square} p & = & \bigcirc \square p & \widehat{\boxminus} p & = & \circledcirc \boxminus p \\
p \Rightarrow q & = & \square(p \rightarrow q) & & & \\
p \Leftrightarrow q & = & \square(p \leftrightarrow q) & & & \\
\exists x : p(x) & = & \neg \forall x : \neg p(x) & & &
\end{array}
$$

A formula that contains no temporal operators is called a *propositional assertion* or simply an assertion.

2.1 Semantics

A *state* s is an interpretation of the variables in $\mathcal{V}$, assigning to each variable $x \in \mathcal{V}$ a truth value. We denote by $s[x] \in \{0, 1\}$ the value assigned to x by state s. We assume that all states interprete F as 0 and T as 1. In the following we use T and F interchangeably for both formulas and truth values. A *model* is an infinite sequence of states:

$$\sigma: \quad s_0, s_1, s_2, \ldots$$

Given a model σ and a temporal formula p, we present an inductive definition for the notion of p holding at a position $j \geq 0$ in σ, denoted by

$$(\sigma, j) \ \models \ p.$$

- For a variable $x \in \mathcal{V}$,

$$(\sigma, j) \models x \quad \Longleftrightarrow \quad s_j[x] = \mathrm{T}$$

For the boolean connectives,

- $(\sigma, j) \models \neg p \quad \Longleftrightarrow \quad (\sigma, j) \not\models p$, i.e., not $(\sigma, j) \models p$
- $(\sigma, j) \models p \vee q \quad \Longleftrightarrow \quad (\sigma, j) \models p$ or $(\sigma, j) \models q$

For the temporal operators,

- $(\sigma, j) \models \bigcirc p \quad \Longleftrightarrow \quad (\sigma, j+1) \models p$
- $(\sigma, j) \models p \mathcal{U} q \quad \Longleftrightarrow$ for some k, $k \geq j$, $(\sigma, k) \models q$, and for every i, $j \leq i < k$, $(\sigma, i) \models p$
- $(\sigma, j) \models \circledcirc p \quad \Longleftrightarrow \quad j = 0$ or $j > 0$ and $(\sigma, j-1) \models p$
- $(\sigma, j) \models p \mathcal{S} q \quad \Longleftrightarrow$ for some k, $0 \leq k \leq j$, $(\sigma, k) \models q$, and for every i, $k < i \leq j$, $(\sigma, i) \models p$

For a variable $x \in \mathcal{V}$, the model σ': $s'_0, s'_1, \ldots$ is said to be an *x-variant* of model σ: $s_0, s_1, \ldots$ if, for each $i = 0, 1, \ldots$, state s'_i agrees with s_i on the interpretation of all variables, except possibly on the interpretation of x.

For the quantifier $\forall$,

- $(\sigma, j) \models \forall x : p \iff (\sigma', j) \models p$, for every σ', an x-variant of σ

For a formula p and a position $j \geq 0$ such that $(\sigma, j) \models p$, we say that p *holds at position* j of σ. If $(\sigma, 0) \models p$, we say that p *holds* on σ, and denote it by $\sigma \models p$.

A useful past formula is the formula *first*, defined as

$$\mathit{first}: \circledcirc \textsc{f}.$$

This formula characterizes the first position in any model. That is, it is false at all positions $j > 0$ and true for $j = 0$.

A temporal formula p is called *satisfiable* if it holds on some model. It is called *valid*, denoted $\models p$, if it holds on all models. Formulas p and q are defined to be *equivalent*, denoted $p \sim q$, if the formula $p \leftrightarrow q$ is valid. Formulas p and q are defined to be *congruent*, denoted $p \approx q$, if the formula $\Box(p \leftrightarrow q)$ (equivalently, $p \Leftrightarrow q$) is valid. Note that congruence is a stronger relation then equivalence. For example, *first* and $\textsc{t}$ are equivalent because both hold at the first position of every model. Obviously, *first* and $\textsc{t}$ are not congruent because *first* does not hold at the second position of any model, but $\textsc{t}$ does.

3 The Proof System

The axiomatic system for QPTL is presented in table 1. Axioms **F**0–**F**7 deal with the future operators, while axioms **P**1–**P**5 deal with the past. Axiom **F**5 represents the *induction* principle. Axiom **M**8 is a *mixed axiom*, containing both future and past operators.

Axioms **Q**1 and **Q**2 deal with the quantifiers. Axiom **Q**1 states that $\forall$ commutes with $\bigcirc$. Axiom **Q**2 stipulates that φ be admissible for $p(x)$, which we take to mean that the sets of variables in φ and $p(x)$ are disjoint. For inference rules, we take TAU, **MP** and $\forall$-**GEN**.

The system consisting of axioms **F**0–**F**7, **P**1–**P**5, the mixed axiom **M**8 and rules **TAU** and **MP**, is shown in [LP00] to be complete for propositional temporal logic (PTL), i.e., the unquantified fragment of QPTL. This is based on [LPZ85] and [Lic91] with a modification of the proof from the *floating* notion of temporal validity to the *anchored* notion of validity. We use this partial completeness result to simplify major portions of our completeness proof, by assuming that every valid PTL formula can be proven by the axiomatic system presented in [LP00].

The completeness proof uses many theorems and derived inference rules. Here we list only some of them. For example, the following theorems can be proven:

	$\neg\exists x: \ p$	$\Leftrightarrow$	$\forall x: \ \neg p$	
	$\neg\forall x: \ p$	$\Leftrightarrow$	$\exists x: \ \neg p$	
$\exists - \textbf{COM} - \ominus$	$\exists x: \ominus p$	$\Leftrightarrow$	$\ominus \exists x: p$	
QT	$p(\varphi)$	$\Rightarrow$	$\exists x: \ p(x)$,	provided φ is admissible for x in $p(x)$.
NFX5	$(\circledcirc \varphi \Rightarrow \varphi)$	$\rightarrow$	$\Box \varphi$	

The last theorem is a derived version of the induction axiom.

Axiom **Q**1 states that $\forall$ commutes with $\bigcirc$. Additional theorems claim that $\forall$ commutes with $\Box$, $\circledcirc$, $\ominus$, and $\boxminus$; and that $\exists$ commutes with $\bigcirc$, $\Diamond$, $\circledcirc$, $\ominus$, and $\diamondminus$.

Axiomatic System **Qx**

Axioms:

F0.	$\Box p \rightarrow p$		
F1.	$\bigcirc \neg p \Leftrightarrow \neg \bigcirc p$	P1.	$\ominus p \Rightarrow \circledcirc p$
F2.	$\bigcirc(p \rightarrow q) \Rightarrow (\bigcirc p \rightarrow \bigcirc q)$	P2.	$\circledcirc(p \rightarrow q) \Rightarrow (\circledcirc p \rightarrow \circledcirc q)$
F3.	$\Box(p \rightarrow q) \Rightarrow (\Box p \rightarrow \Box q)$	P3.	$\Box p \rightarrow \Box \circledcirc p$
F4.	$\Box p \rightarrow \Box \bigcirc p$	P4.	$(p \mathcal{S} q) \Leftrightarrow [q \vee (p \wedge \ominus(p \mathcal{S} q))]$
F5.	$(p \Rightarrow \bigcirc p) \rightarrow (p \Rightarrow \Box p)$	P5.	$\circledcirc$ F
F6.	$(p \mathcal{U} q) \Leftrightarrow [q \vee (p \wedge \bigcirc(p \mathcal{U} q))]$		
F7.	$p \mathcal{U} q \Rightarrow \Diamond q$	M8.	$\varphi \Rightarrow \bigcirc \ominus \varphi$
Q1.	$\forall x : \bigcirc p \Leftrightarrow \bigcirc \forall x : p$		
Q2.	$\forall x : p(x) \Rightarrow p(\varphi)$, where φ is a formula admissible for x in $p(x)$.		

Inference Rules:

TAU.	for every p, a temporal instantiation of a propositional tautology, $\vdash \Box p$
MP.	$p \rightarrow q, \quad p \vdash q$
$\forall$-GEN.	$p \Rightarrow q(x) \vdash p \Rightarrow \forall x : q(x)$, provided p does not refer to x.

Table 1: The axiomatic system for QPTL.

Some of the derived inference rules are:

INST		$p(x)$	$\vdash p(\varphi)$, provided φ is admissible for x in $p(x)$
$\exists$-INTR		$p(x) \Rightarrow q$	$\vdash \exists x : p(x) \Rightarrow q$, provided x does not occur in q
$\forall$-INTR		$p \Rightarrow q(x)$	$\vdash p \Rightarrow \forall x : q(x)$, provided x does not occur in p
$\forall\forall$-INTR	$p(x) \Rightarrow q(x)$		$\vdash \forall x : p(x) \Rightarrow \forall x : q(x)$
	$p(x) \Leftrightarrow q(x)$		$\vdash \forall x : p(x) \Leftrightarrow \forall x : q(x)$
$\exists\exists$-INTR	$p(x) \Rightarrow q(x)$		$\vdash \exists x : p(x) \Rightarrow \exists x : q(x)$
	$p(x) \Leftrightarrow q(x)$		$\vdash \exists x : p(x) \Leftrightarrow \exists x : q(x)$
SUBST		$p \Leftrightarrow q$	$\vdash \varphi(p) \Leftrightarrow \varphi(q)$, provided p and q are admissible for x in $\varphi(x)$.

Note that it is congruence, rather than equivalence, which supports "substitution of equals for equals."

To abbreviate some of our detailed proofs, we sometimes group several steps into a single step, using the following justifications.

- **PR** - *Propositional reasoning.* If the current proof line can be established from the previous proof lines $n_1, \ldots, n_k$, using propositional reasoning, we affix the justification **PR**, $n_1, \ldots, n_k$

to the current line. Let $p_1, \ldots, p_k$ be the formulas at lines $n_1, \ldots, n_k$ and q be the formula at the current line. The inference of q from $p_1, \ldots, p_k$ by **PR** implies that the formula

$$p_1 \wedge p_2 \wedge \ldots \wedge p_k \rightarrow q$$

is a propositional tautology or a temporal instantiation of a propositional tautology in which some propositions have been replaced by QPTL formulas.

- **TR** - *Temporal Reasoning.* A proof line q can be derived from lines $p_1, \ldots, p_k$ by the **TR** proof principle if

  $$p_1 \wedge \ldots \wedge p_k \rightarrow q$$

  is a (quantifier free) temporal propositional tautology or an instantiation of a temporal tautology in which some propositions have been replaced by QPTL formulas. By [LP00], all temporal propositional tautologies can be derived within **Qx** which is a superset of the axiomatic system used in [LP00].

- **QR** - *Quantifier Reasoning.* We use this proof principle to abbreviate proof steps which involve simple and obvious applications of the quantifier axioms **Q**1, **Q**2 and the quantifier rule $\forall$-**GEN**.

Let φ be a QPTL formula. We use the notation

$$\vdash \varphi$$

to denote that φ is provable within the axiomatic system **Qx**.

Claim 1 *(Soundness) The axiomatic system* **Qx** *is sound. That is, if* $\vdash \varphi$ *then* $\models \varphi$.

The claim can be established by showing that all the axioms in **Qx** are valid, and that the inference rules infer valid consequence from valid premises.

4 History and Prophecy Schemes

In this section, we establish several theorems leading to the proofs of the history and prophecy schemes, within the proof system **Qx**.

As explained in the introduction, one of our motivations for the study of QPTL was to provide a logic-oriented formulation of the complete method proposed in [AL91] for proving that one system refines another. The notions of history and prophecy schemes are central to their proof method and can be shown to correspond, respectively, to the notions of *forward* and *backward* simulation which were previously used for establishing refinement relations between two reactive systems. In our preliminary studies we conjectured that it will be necessary to introduce two axioms which will represent these two fundamental concepts. Later, we realized that these two

concepts can actually be established as theorems of our proof system. These theorems are the topic of this section.

Both schemes use a recursive definition of a function. The history (forward) scheme can be represented by the recursive definition $y = f(\ominus y)$[1] which defines the current value of the temporal variable y in terms of its previous value. In a symmetric way, the prophecy (backward) scheme can be represented by $y = g(\bigcirc y)$, defining the current value of y in terms of its next value. The scheme theorems state that for any f and g the two schemes have a solution. This can be represented by the two theorems:

History	$\exists y : \Box(y = f(\ominus y))$
Prophecy	$\exists y : \Box(y = g(\bigcirc y))$

4.1 Variables over Fixed Finite Domains

The basic logic introduced above allows only boolean variables. In order to express the behaviors of automata, it is convenient to consider variables which range over an arbitrary fixed finite domain D. Without loss of generality, we can take D to be a segment of the integers $\{1, \ldots, n\}$ where we may assume that $n = 2^r$ for some $r > 0$. Consequently, we may extend the basic logical language by a vocabulary V_D of *D-variables*, *D-expressions*, and *D-constants*: $\mathcal{K}$: $\{k_1, \ldots, k_n\}$. To the syntactical definition of QPTL, we add the following clauses:

- Every D-variable $y \in V_D$ and every D-constant $k \in \mathcal{K}$ are D-expressions.
- If e is a D-expression, then so are $\bigcirc e$ and $\ominus e$, intended to represent the value of e at the next and previous positions of the model, respectively. If $\ominus e$ is evaluated at position 0, it yields the default value k_1.
- If p is a formula and e_1 and e_2 are (D)-expressions, then

  if p **then** e_1 **else** e_2

  is an expression whose intended meaning is that it evaluates to the value of e_1 if p holds, otherwise, its value is that of e_2.
- If e_1 and e_2 are expressions, then $e_1 = e_2$ is an (atomic) formula.

This generalization is not a real extension since D-variables, D-expressions and D-constants can be encoded by r-tuples of boolean variables, r-tuples of boolean formulas, and r-tuples of the boolean constants, ranging over T and F. This is done as follows:

The D-variable y can be encoded by the boolean variables $x_1, \ldots, x_r$, each representing a bit in the binary representation of y. An expression will be represented by a tuple of formulas $\varphi_1, \ldots, \varphi_r$. For example, if $\varphi_1, \ldots, \varphi_r$ is the tuple representation of expression e_1 and $\psi_1, \ldots, \psi_r$ is the tuple

[1] Here and until the end of this paragraph, a scheme such as $y = f(\ominus y)$ is an abbreviation for the conjunction $y_1 \leftrightarrow f_1(\ominus y_1, \ldots, \ominus y_n) \wedge \ldots \wedge y_r \leftrightarrow f_r(\ominus y_1, \ldots, \ominus y_r)$, where $y_1, \ldots, y_r$ are boolean variables. A more precise definition of this shorthand notation is given in the next subsection.

representation of expression e_2, then the tuple representations of the expressions $\bigcirc e_1$, $\ominus e_1$ and **if** p **then** e_1 **else** e_2, and the representation of the formula $e_1 = e_2$ are given by:

$$\begin{array}{rcl} \bigcirc e_1 & -- & (\bigcirc \varphi_1, \ldots, \bigcirc \varphi_r) \\ \ominus e_1 & -- & (\ominus \varphi_1, \ldots, \ominus \varphi_r) \\ \textbf{if } p \textbf{ then } e_1 \textbf{ else } e_2 & -- & ((p \wedge \varphi_1) \vee (\neg p \wedge \psi_1), \ldots, (p \wedge \varphi_r) \vee (\neg p \wedge \psi_r)) \\ e_1 = e_2 & -- & \varphi_1 \leftrightarrow \psi_1 \wedge \cdots \wedge \varphi_r \leftrightarrow \psi_r \end{array}$$

It is not difficult to see that all the axioms, theorems, and rules presented in section 3 can be extended to cover D-variables and expressions. We refer to expressions also as *functions*.

The two-branch conditional statement **if** p **then** e_1 **else** e_2 can be extended to an $m+1$-branch conditional as follows:

Let $p_1, \ldots, p_m$ be QPTL formulas and $e_1, \ldots, e_{m+1}$ be expressions. Then

$$\textbf{if } p_1 \textbf{ then } e_1 \textbf{ else-if } p_2 \textbf{ then } e_2 \textbf{ else-if } \ldots \textbf{ else-if } p_m \textbf{ then } e_m \textbf{ else } e_{m+1}$$

is an $m+1$-branch conditional expression. It can be defined as an $m+1$ nesting of two-branch conditionals.

We let $min\, u\,.\, \varphi(u)$ be a shorthand notation for

$$\begin{array}{ll} \textbf{if} & \exists u : \varphi(u) \wedge u = k_1 \textbf{ then } k_1 \\ \textbf{else-if} & \exists u : \varphi(u) \wedge u = k_2 \textbf{ then } k_2 \\ \quad . & \\ \quad . & \\ \textbf{else-if} & \exists u : \varphi(u) \wedge u = k_n \textbf{ then } k_n \\ \textbf{else} & k_1 \end{array}$$

Note the convention by which the scope of quantifiers extends as far to the right as possible. Obviously, if there exists a u such that $\varphi(u)$ holds, then $min\, u\,.\, \varphi(u)$ yields the constant k_i with the smallest index such that $\varphi(u) \wedge u = k_i$ holds. If there does not exist such a u, then $min\, u\,.\, \varphi(u)$ return the default value k_1. Note that minimization is only applied to the value of u at the *current* position, but u may have arbitrary different values at different positions.

4.2 Extensible Formulas

A QPTL formula $\varphi(y)$ is called $(y-)$*past dependent* if

$$\vdash \boxminus(y = z) \Rightarrow \varphi(y) \leftrightarrow \varphi(z)$$

Thus, a $(y-)$past-dependent formula $\varphi(y)$, depends only on the past and present values of variable y, and this restricted dependency is provable in $\mathbf{Q_X}$. Note that $\varphi(y)$ may depend on both past and future values of any other variables which are free in $\varphi(y)$.

A past-dependent formula $\varphi(y)$ is called *extensible* if

$$\vdash \circledcirc \varphi(y) \Rightarrow \exists z : \varphi(z) \wedge \widehat{\boxminus}(y = z)$$

This provable entailment requires that if $\varphi(y)$ holds at the previous position, we can always find a sequence, represented by z, which agrees with y on all preceding positions and causes φ to hold at the present position. It allows provability of the validity of φ to be extended from the previous to the current position, by appropriate extension of y into z.

An extensible formula $\varphi(y)$ is called *uniquely extensible* if both

$$\vdash \varphi(y) \wedge \varphi(z) \Rightarrow (y = z) \quad \text{and} \quad \vdash \varphi(y) \Leftrightarrow \boxminus \varphi(y)$$

The first formula states that $\varphi(y)$ uniquely determines the value of y at the current position. The second formula implies that if φ holds now, it must have held at all preceding positions. Unique extensibility requires the provability of both these properties. Thus, it is provable in $\mathbf{Q_X}$ that a uniquely extensible $\varphi(y)$ uniquely determines the value of y at the current and all preceding positions.

The proofs of the following Lemmas, Claims and Theorems can be found in the appendix.

Lemma 2 *Let $\varphi(y)$ be a* QPTL *formula. Then*

$$\vdash \varphi(y) \Rightarrow \exists z : (\varphi(z) \wedge z = \mathit{min}\ u\,.\,\varphi(u))$$

This lemma claims that the following is provable in $\mathbf{Q_X}$: If φ is satisfied by some y then it is also satisfied by some z whose current value (value at the current position) is minimal. The lemma gives a closed form expression for the value of z at the current position, but not at any other position.

To illustrate the relevant concepts, consider the formula

$$\varphi'(y) : y \vee \widehat{\boxminus} y$$

Clearly, $\varphi'(y)$ holds at position j if either $y[j] = \mathrm{T}$ or $y[k] = \mathrm{T}$, for all k, $0 \leq k < j$. Formula $\varphi'(u)$ is extensible, because given any y and position $j \geq 0$, we can always pick a z such that $z[j] = \mathrm{T}$ and $z[k] = y[k]$ for all k, $0 \leq k < j$. Such a z will satisfy φ' at position j.

Lemma 1 holds for φ'. First, we observe that $\min u\,.\,\varphi'(u) = \mathrm{F}$ (assuming the ordering $\mathrm{F} < \mathrm{T}$). This is because u which is F at position j and T elsewhere satisfies $\varphi'(u)$ and has the minimal value F at position j. We can take $z = u$ (at all positions) to satisfy Lemma 1.

Note that φ' is *not* uniquely extensible, because $\varphi'(\mathrm{F})$ and $\varphi'(\mathrm{T})$ both hold at position 0, yet $\mathrm{F} \neq \mathrm{T}$.

Lemma 3 *Let $\varphi(y)$ be an extensible formula. Then*

$$\vdash \Box\, \exists y : \varphi(y)$$

This lemma claims that, for every position j, we can provably find a sequence y such that $\varphi(y)$ holds at j.

Claim 4 *Let $\varphi(y)$ be a uniquely extensible formula. Then*

$$\vdash \varphi(y) \Rightarrow (y = \mathit{min}\ u\,.\,\varphi(u))$$

This claim states that at any position where $\varphi(y)$ holds, y has provably a locally minimal value among all sequences satisfying φ. This is not surprising, since there is provably only one unique y (over all past positions) that can satisfy φ, due to the unique extensibility of φ.

Claim 5 *Let $\varphi(y)$ be uniquely extensible. Then*

$$\vdash \varphi(y) \Rightarrow \varphi(\mathit{min}\ u\,.\,\varphi(u))$$

This claim states that if φ is satisfiable, it is provably satisfiable by a sequence y which is locally minimal among all φ-satisfying sequences, *at all positions*. This follows from unique extensibility, claiming that at any position, there is only one way to extend y to satisfy φ.

The assumption of unique extensibility is essential for Claim 5. Consider the formula

$$\varphi' : y \vee \widehat{\boxminus} y$$

which has been shown to be extensible but not uniquely extensible. As shown above, $\min u\,.\,\varphi'(u) = \textsc{f}$. However, $\varphi'(\textsc{f})$ holds at no position, except for position 0.

The importance of Claim 5 is not so much in that we found a sequence satisfying φ but in the fact that we have a syntactic representation $\mathit{min}\ u\,.\,\varphi(u)$ for this satisfying sequence. Once we have a closed-form expression for this, we can apply theorem **QT** as is done in the following Lemma.

Lemma 6 *Let $\varphi(y)$ be uniquely extensible. Then*

$$\vdash \exists y :\ \Box\,\varphi(y)$$

This lemma combines Lemma 3, Claim 5, and Theorem **QT**, to obtain the provable existence of y satisfying φ at *all* positions.

Let $\varphi(y)$ be extensible. Define

$$\mathit{sext}_\varphi(y) : \varphi(y) \wedge (y = \mathit{min}\ u\,.\,(\varphi(u) \wedge \widehat{\boxminus}(u = y)))$$

This formula requires a y that satisfies φ and that is locally minimal among all sequences satisfying φ and agreeing with y on all preceding positions. The idea is to convert an arbitrary extensible formula into a uniquely extensible one, as shown in the following lemma:

Lemma 7 *Let $\varphi(y)$ be an extensible formula. Then*

$$\boxminus\ \mathit{sext}_\varphi(y)$$

is provably a uniquely extensible formula.

Lemma 8 *Let $\varphi(y)$ be an extensible formula. Then*

$$\vdash \exists y :\ \Box\,\varphi(y)$$

This theorem claims that if φ is an *extensible* formula, then it is provable in **Qx** that there exists a sequence y satisfying φ at all positions. It extends the statement made in lemma 6 for the more restricted class of *uniquely-extensible* formulas. Lemma 8, together with claims 9 and 10 below, are the basis for showing provability of both the history and the prophecy schemes, as shown in the next subsection.

Let $step(t)$ represent the following QPTL formula

$$step(t) \quad : \quad \boxminus t \wedge \widehat{\Box}\neg t.$$

The formula $step(t)$ characterizes t as a *step* function, being true up to (and including) the present position, and false from the next position on. Such sequences are used to mark the current position.

Claim 9 $\quad \vdash \Box\, \exists t : step(t)$

This claim states that, for every position j, there provably exists a step function marking the position.

Let g_1, g_2, and g_3 be three expressions, and z be a variable that does not occur free in any of these expressions. Then,

Claim 10 $\quad \vdash \Box\, \exists z : (\widehat{\boxminus}(z = g_1) \wedge (z = g_2) \wedge \widehat{\Box}(z = g_3))$

This claim states the following provability. We can always patch together three expressions and claim the existence of a sequence z whose value equals g_1 at all positions preceding the current one, z equals g_2 at the current position, and z equals g_3 at all positions succeeding the current position.

4.3 The History Scheme

A function (D-expression) $h(y)$ is called *historic* if it satisfies

$$\vdash \widehat{\boxminus}(y = z) \Rightarrow (h(y) = h(z))$$

Namely, $h(y)$ is historic if it is provable that the value of $h(y)$ depends only on the value of y at the preceding positions. Note that our definition extends that of Abadi and Lamport in that it allows h at position j to depend on the value of y at *all* preceding positions $k < j$, rather than just the value of y at $j - 1$. Another generalization is that h may depend on values of variables other than y at *all* positions (including $k \geq j$).

Lemma 11 *Let $h(y)$ be a historic function. Then*

$$\varphi(y) : y = h(y)$$

is an extensible formula.

Theorem 1 *Let $h(y)$ be a historic function and $\varphi(y) : y = h(y)$. Then*

$$\vdash \exists y : \Box\, \varphi(y)$$

This theorem establishes that a history scheme always defines some sequence y satisfying the recurrence equation $y = h(y)$ at all positions, and that this fact is provable in **Qx**.

4.4 The Prophecy Scheme

A function (D- expression) $f(y)$ is called *prophetic* if it satisfies

$$\vdash \bigcirc(y = z) \Rightarrow (f(y) = f(z))$$

Namely, $f(y)$ is prophetic if it is provable that the value of $f(y)$ depends only on the value of y at the next position. Note that our definition extends that of Abadi and Lamport in that it allows f to depend on values of variables other than y at *all* positions.

Let $\varphi(y) : y = f(y)$, where $f(y)$ is a prophetic function. We define the following QPTL formula:

$$\psi(y) : \exists t : \mathit{step}(t) \wedge \Box \Diamond \exists u : (\boxminus \varphi(u) \wedge \boxminus(t \rightarrow u = y)) \tag{5}$$

This formula lies at the heart of the reason why prophecy schemes, which are based on backward induction over a finite domain, identify a well-defined sequence y satisfying $y = f(y)$ at all positions. The formula uses the step variable t to mark the current position. Then it requires the existence of infinitely many future points, at which we can start a backwards induction on a sequence u (that may vary from one future starting point to another), requiring that all these sequences when they come back to the current position (and below) agree with the current value of y. The informal proof that a prophecy scheme has a solution is based on the Ramsey theorem, presented here in a simplified form.

Theorem 2 *For every finite partition of the natural numbers*

$$P_1 \cup \ldots \cup P_h = \mathsf{N},$$

some partition P_i, $i \in [1..h]$, is infinite.

The formula $\psi(y)$ gives Theorem 2 a formal expression, as can be seen by the following interpretation. Let h be the size of the domain of variable y. Namely, $v_1, \ldots, v_h$ are the possible values of y, at each position. Let P_i, where $1 \leq i \leq h$, be the set of all future positions $k \in \mathsf{N}$ such that starting a backwards induction on a sequence u at position k, results in $y = v_i$ at the current position. Following Theorem 2, for some i, $1 \leq i \leq h$, P_i is guaranteed to be an infinite set, explaining the use of $\Box \Diamond \exists u$ (and not $\Box \exists u$) in the definition of $\psi(y)$.

Theorem 2 relies on the partition being finite, and the proof of existence of a well-defined sequence y satisfying $y = f(y)$ indeed relies on the domain of y being finite, as can be seen in the proof of Lemma 15 which is the basis for the main theorem (Theorem 3).

Claim 12 *Let $f(y)$ be a prophetic function. Then*

$$\vdash \boxminus(y = z) \Rightarrow \widehat{\boxminus}(f(y) = f(z))$$

Claim 13 *Let $f(u)$ be a prophetic function. Then*

$$\vdash \Box \forall v \, \exists u : \bigcirc(u = v) \wedge \boxminus(u = f(u))$$

This claim states the following provability. Given an *initial value* v, we can always apply the backward induction and obtain a sequence u satisfying $u = f(u)$ at the current and all preceding positions, and whose value at the next position equals v.

Corollary 14 $\vdash \Box \exists u : \boxminus(u = f(u))$

This corollary claims that at every position, one can find a sequence u satisfying $u = f(u)$ at the present and all preceding positions, and that this fact is provable.

Lemma 15 *Let $f(y)$ be a prophetic function and $\varphi(y) : y = f(y)$. Then the formula ψ as defined in (5) is provably an extensible formula.*

This lemma establishes that the formula ψ as defined in (5), is provably an extensible formula. The extensibility of ψ together with Lemma 8 are used to show the provability of existence of a sequence y satisfying ψ at all positions, namely $\vdash \exists y : \Box \psi(y)$. This together with the next lemma (Lemma 16), establishes the provability of existence of prophecy schemes. Note that the proof of Lemma 15 relies on the finiteness of y's domain.

Lemma 16 *Let $f(y)$ be a prophetic function and $\varphi(y) : y = f(y)$. Then*

$$\vdash \psi(y) \Rightarrow \widehat{\boxminus}\varphi(y)$$

This lemma claims that if ψ holds at some position, then $y = f(y)$ holds at all preceding positions, and that this fact is provable.

Theorem 3 *Let $f(y)$ be a prophetic function and $\varphi(y) : y = f(y)$. Then*

$$\vdash \exists y : \Box \varphi(y)$$

This theorem establishes that a prophecy scheme always defines some sequence y satisfying the recurrence equation $y = f(y)$ at all positions, and that this fact is provable in **Qx**.

5 From QPTL to ω-Automata

5.1 Basic Definitions

An ω-automaton $\mathcal{A} = (Q, Q_0, \delta, C)$ consists of

- Q – a finite set of automaton-locations[2].
- $Q_0 \subseteq Q$ – a subset of *initial* automaton-locations.
- δ – For every $q_i, q_j \in Q$, $\delta(q_i, q_j)$ is a propositional assertion over $\mathcal{V}$, a countable set of boolean variables (i.e., a formula containing no quantifiers and no temporal operators).
- C – an acceptance condition.

[2]We use the term *location* instead of the common term *state*, to avoid the confusion with model-states.

Let $\sigma : s_0, s_1, \ldots$ be a model. A sequence of automaton-locations $\rho : q_0, q_1, \ldots$ is a *run-segment* of $\mathcal{A}$ over σ, if

$$s_i \models \delta(q_i, q_{i+1}),$$

for every $i \geq 0$. A run-segment $\rho : q_0, q_1, \ldots$ is a *run* of $\mathcal{A}$ if $q_0 \in Q_0$. The *infinity set* $inf(\rho)$ of a run ρ is the set of automaton-locations that are visited infinitely many times in ρ. A run is *accepting* if the infinity set $inf(\rho)$ satisfies the acceptance condition C. A model σ is said to be *accepted* by the automaton $\mathcal{A}$, if $\mathcal{A}$ has an accepting run over σ. We denote by $\mathcal{L}(\mathcal{A})$ (the *language* of $\mathcal{A}$), the set of all models accepted by $\mathcal{A}$. Two automata $\mathcal{A}_1$ and $\mathcal{A}_2$ are called *equivalent* if $\mathcal{L}(\mathcal{A}_1) = \mathcal{L}(\mathcal{A}_2)$, namely, for every model σ, σ is accepted by $\mathcal{A}_1$ iff σ is accepted by $\mathcal{A}_2$.

Classes of ω-automata are defined according to their acceptance conditions. We denote by B, R and S the ω-automata with Buchi, Rabin, and Streett acceptance conditions, respectively. The acceptance conditions for these three classes are summarized in the table below.

	Syntax	Semantics
B	$F \subseteq Q$	$inf(\rho) \cap F \neq \emptyset$
R	$\bigvee_i (L_i \wedge \neg U_i)$, where $L_i, U_i \subseteq Q$	$\exists i: \quad inf(\rho) \cap L_i \neq \emptyset \ \wedge \ inf(\rho) \cap U_i = \emptyset$
S	$\bigwedge_i (L_i \rightarrow U_i)$, where $L_i, U_i \subseteq Q$	$\forall i: \quad inf(\rho) \cap L_i = \emptyset \ \vee \ inf(\rho) \cap U_i \neq \emptyset$

A model σ' is said to be a *j-marked variant* of σ if σ' is a t-variant of σ and σ' interprets t as T at position j and F elsewhere. Observe that every model σ has a unique j-marked variant for each $j \geq 0$.

Automaton $\mathcal{A}$ *j-approves* a model σ if it accepts the j-marked variant of σ.

Let φ be a QPTL formula not referring to the boolean variable t, and $\mathcal{A}$ be an ω-automaton. We say that $\mathcal{A}$ is an *automaton congruent to the formula* φ if, for every model σ and position $j \geq 0$, $(\sigma, j) \models \varphi$ iff $\mathcal{A}$ j-approves σ. Note that this definition overloads the notion of congruence which may now hold between two formulas as well as between an automaton and a formula. This extension is, however, consistent in the sense that if automaton $\mathcal{A}$ is congruent to formula φ and also to formula ψ then φ and ψ are congruent formulas.

Let $\mathcal{A} = (Q, Q_0, \delta, C)$ be an ω-automaton. We say that an *automaton-location* $q \in Q$ is *deterministic* if for every model s, $\mid \{q' \mid s \models \delta(q, q')\} \mid \leq 1$. We say that the *automaton* $\mathcal{A}$ is deterministic if all its locations are, namely, for all $q \in Q$, q is deterministic.

Let $\mathcal{A} = (Q, Q_0, \delta, F)$ be a Büchi automaton, and $q, q' \in Q$. We say that q' is *reachable* from q in $\mathcal{A}$ if there exists a model σ and a sequence of locations $\rho : q_0 = q, q_1, \ldots, q_j = q', \ldots$ such that ρ is a run-segment of $\mathcal{A}$ over σ. We say that $\mathcal{A}$ is *deterministic-in-the-limit* if for every location $q' \in Q$ such that q' is reachable from an accepting location $q \in F$, q' is deterministic. This implies that every accepting run can contain only finitely many non-deterministic automata locations. We use the notation D, N and L for deterministic, non-deterministic and deterministic-in-the-limit automata, respectively.

Let $\mathcal{A}$ be an ω-automaton. We say that $\mathcal{A}$ is *void* if there is no model σ and no position $j \geq 0$, such that σ is j-approved by $\mathcal{A}$. We say that $\mathcal{A}$ is *initially void* if there is no model σ which is 0-approved by $\mathcal{A}$. Note that an automaton which only accepts the empty language is necessarily

void. On the other hand, an automaton may be void and yet accept a non-empty language. However, all the models it accepts are not a j-marked variant of some model, meaning that the number of positions at which t is true, in any of the accepted models, is different from 1.

For every automaton $\mathcal{A}$, we construct a QPTL formula $\chi_{\mathcal{A}}$, characterizing the approving runs of $\mathcal{A}$. The formula uses a variable y which ranges over Q to denote the location in which the automaton is currently situated. We write $at\_q_i(y)$ as a synonym for $y = q_i$ and, for a set of locations $S \subseteq Q$, we write $in\_S(y)$ as a synonym for $\bigvee_{q_i \in S} at\_q_i(y)$. The formula is defined in several stages as follows:

$$
\begin{array}{lcl}
S(t) & : & \widehat{\boxminus}\neg t \wedge t \wedge \widehat{\Box}\neg t \\
init_{\mathcal{A}}(y) & : & in\_Q_0(y) \\
run_{\mathcal{A}}(y) & : & \Box \bigvee_{i,j}(at\_q_i(y) \wedge \bigcirc at\_q_j(y) \wedge \delta(q_i, q_j)) \\
acc\_r_{\mathcal{A}}(y) & : & init_{\mathcal{A}}(y) \wedge run(y)_{\mathcal{A}} \wedge acc_{\mathcal{A}}(y) \\
app\_r_{\mathcal{A}}(y) & : & \diamondminus(first \wedge acc\_r_{\mathcal{A}}(y)) \\
\chi_{\mathcal{A}} & : & \exists t, y \ : \ S(t) \wedge app\_r_{\mathcal{A}}(y)
\end{array}
$$

Formula $S(t)$ characterizes the special variable $t \in \mathcal{V}$ as a proper marker which is true at the current position and false at all other positions. Formula $init_{\mathcal{A}}(y)$ requires that the automaton currently resides at an initial location. Formula $run_{\mathcal{A}}(y)$ requires that, from now on, variable y will follow the transition rules, moving from location q_i to q_j only when $\delta(q_i, q_j)$ holds. Finally, formula $\chi_{\mathcal{A}}$ groups all these components together, requiring that t marks the current position and that, if we go back to the beginning of the model, we can interpret the values of y as encoding an accepting run of the automaton.

Note that, since $\delta(q_i, q_j)$ is an assertion over $\mathcal{V}$, the formulas $run_{\mathcal{A}}(y)$, $acc\_r_{\mathcal{A}}(y)$ and $app\_r_{\mathcal{A}}(y)$ may have any of the variables in $\mathcal{V}$ as free variables, including the special variable t. In the following, we parameterize all these functions with y, but add a second parameter (t or $v \in \mathcal{V} - \{t\}$) only when relevant for a given proof.

The acceptance formula $acc_{\mathcal{A}}(y)$ depends, of course, on the acceptance type. For the three considered types, it is defined as follows:

$$
\begin{array}{lcl}
acc_{\mathcal{A}_{NB}}(y) & : & \Box \diamond in\_F(y) \\
acc_{\mathcal{C}_{DR}}(y) & : & \bigvee_i(\Box \diamond in\_L_i(y) \wedge \diamond \Box \neg in\_U_i(y)) \\
acc_{\mathcal{D}_{DS}}(y) & : & \bigwedge_i(\diamond \Box \neg in\_L_i(y) \vee \Box \diamond in\_U_i(y))
\end{array}
$$

Claim 17 *Every automaton $\mathcal{A}$ is congruent to its characteristic formula $\chi_{\mathcal{A}}$. That is, for every model σ and position $j \geq 0$, $(\sigma, j) \models \chi_{\mathcal{A}}$ iff $\mathcal{A}$ j-approves σ.*

5.2 Complementation of ω-Automata

In subsection 5.3, we show how to construct for each formula φ an NB-automaton which is congruent to φ. The construction is inductive, showing for each operator of the logic how to build an automaton that corresponds to a formula using this operator from the automata corresponding to its operands. Based on this construction, we then show that the congruence $\varphi \Leftrightarrow \chi_{\mathcal{A}_\varphi}$ is provable in the axiomatic system **Qx**. For all but the negation operator, the construction is

straightforward and requires no introduction. The construction for the negation operator is introduced below.

A classical approach to the complementation of Büchi automata is via determinization into automata with stronger acceptance conditions. An optimal determinization construction is given in [Saf88]. Another complementation construction which circumvents the need for determinization, is presented in [Kla91]. A third approach, using a translation to weak alternating automata, is given in [KV97]. All three methods enable a $2^{O(nlogn)}$ complementation construction, matching the known lower bound [Mic88].

Since we use the automata to show provability, we are not concerened with complexity issues, but rather, simplicity and clarity of the construction. We therefore adopt a simplification of [Saf88] construction. Let p be a formula and $\mathcal{A}_p$ be an automata congruent to p. Following ([Saf88]), we construct the sequence

$$\mathcal{A}_p \xrightarrow{step1} \mathcal{C}_{DR} \xrightarrow{step2} \mathcal{D}_{DS} \xrightarrow{step3} \mathcal{A}_{\neg p},$$

where $\mathcal{C}_{DR}$ is a deterministic Rabin automaton equivalent to $\mathcal{A}_p$, $\mathcal{D}_{DS}$ is a deterministic Streett automaton which is the complement of $\mathcal{C}_{DR}$, and $\mathcal{A}_{\neg p}$ is a non-deterministic Büchi automaton equivalent to $\mathcal{D}_{DS}$. The relations holding between these automata are those of equivalence, which is stronger than congruence. Certainly if $\mathcal{A}_1$ and $\mathcal{A}_2$ are equivalent, i.e., accept the same language, then they are congruent. That is, $\mathcal{A}_1$ j-approves model σ iff $\mathcal{A}_2$ j-approves σ.

In te following, we present the first step of the construction, namely the determinization. In [Saf88], determinization is performed in a single step. We use a simplified construction which proceeds in two steps. Starting with a non-deterministic Büchi automaton $\mathcal{A}_{NB}$, we first construct a deterministic-in-the-limit Büchi automaton $\mathcal{B}_{LB}$ which is equivalent to $\mathcal{A}_{NB}$ [CY95]. Next we construct a deterministic automaton $\mathcal{C}_{DR}$ with Rabin acceptance condition, which is equivalent to $\mathcal{B}_{LB}$.

Let $\mathcal{A} = \langle Q, Q_0, \delta, C\rangle$ be an automaton and $S, S' \subseteq Q$. We denote by $\Delta^{\mathcal{A}}(S, S')$ the propositional assertion which represents δ over *sets of locations,* as follows:

$$\Delta^{\mathcal{A}}(S, S') \ : \ \bigwedge_{r \in S'} \bigvee_{q \in S} \delta(q, r) \ \wedge \ \bigwedge_{r \notin S'} \neg \bigvee_{q \in S} \delta(q, r)$$

A state s satisfies $\Delta^{\mathcal{A}}(S, S')$, iff the following two conditions are satisfied:

- For every location $r \in S'$, there exists a location $q \in S$ such that $s \models \delta(q, r)$.
- For every location $r \in Q$ such that $r \notin S'$, there exists no location $q \in S$ such that $s \models \delta(q, r)$.

Let $\sigma : s_0, \ldots, s_m$ be a model-segment. We write $\sigma \models \Delta^{\mathcal{A}}(S, S')$ to denote a sequence of propositional assertions

$$\Delta^{\mathcal{A}}(S_0 = S, S_1), \Delta^{\mathcal{A}}(S_1, S_2), \ldots, \Delta^{\mathcal{A}}(S_m, S')$$

satisfying $s_i \models \Delta^{\mathcal{A}}(S_i, S_{i+1})$, for every $i \in [0..m]$.

Let $\mathcal{A}_{NB} = \langle Q^{\mathcal{A}}, Q_0, \delta^{\mathcal{A}}, F^{\mathcal{A}} \rangle$ be a non-deterministic Büchi automaton. We define a deterministic-in-the-limit automaton $\mathcal{B}_{LB} = \langle Q^{\mathcal{B}}, Q_0, \delta^{\mathcal{B}}, F^{\mathcal{B}} \rangle$ as follows:

$$Q^{\mathcal{B}} \quad = \quad Q^{\mathcal{A}} \cup \{(V, W) \mid V, W \subseteq Q^{\mathcal{A}},\ V \cap W = \emptyset,\ V \cap F^{\mathcal{A}} = \emptyset,\ V \cup W \neq \emptyset\}$$

The transition assertion $\delta^{\mathcal{B}}$ is defined as follows:
For $q, q' \in Q^{\mathcal{A}} \cap Q^{\mathcal{B}}$ such that $q' \notin F^{\mathcal{A}}$, then

$$\delta^{\mathcal{B}}(q, q') = \delta^{\mathcal{A}}(q, q').$$

For $q, q' \in Q^{\mathcal{A}} \cap Q^{\mathcal{B}}$ such that $q' \in F^{\mathcal{A}}$, then both

$$\delta^{\mathcal{B}}(q, q') = \delta^{\mathcal{A}}(q, q') \text{ and } \delta^{\mathcal{B}}(q, (\emptyset, \{q'\})) = \delta^{\mathcal{A}}(q, q').$$

For $(V_1, W_1), (V_2, W_2) \in Q^{\mathcal{B}} - Q^{\mathcal{A}}$,

$$\delta^{\mathcal{B}}((V_1, W_1), (V_2, W_2)) = \begin{cases} \text{If} \quad V_1 = \emptyset \text{ and } W_2 \subseteq F^{\mathcal{A}} & \text{then} \quad \Delta^{\mathcal{A}}(W_1, V_2 \cup W_2) \\ \text{Else if} \quad V_1 = \emptyset \text{ and } W_2 \not\subseteq F^{\mathcal{A}} & \text{then} \quad \text{F} \\ \text{Otherwise} \quad \Delta^{\mathcal{A}}(V_1 \cup W_1, V_2 \cup W_2) \wedge \bigwedge_{r \in V_2} \bigvee_{q \in V_1} \delta^{\mathcal{A}}(q, r) \\ \qquad \wedge \bigwedge_{r \in (W_2 - F^{\mathcal{A}})} \bigvee_{q \in W_1} \delta^{\mathcal{A}}(q, r) \end{cases}$$

Thus, for the case that $V_1 = \emptyset$, we place in V_2 all the non-accepting successors of W_1, and place in W_2 all the accepting successors.

For $V_1 \neq \emptyset$, we place in V_2 all the non-accepting successors of V_1 except those which also belong to the successors of W_1. The set W_2 gets all the successors of W_1 plus the accepting successors of V_1.
The accepting set of $\mathcal{B}_{LB}$:

$$F^{\mathcal{B}} \quad = \quad \{(\emptyset, W) \mid W \subseteq Q^{\mathcal{A}}\}$$

The following Claim states that the two automata $\mathcal{A}_{NB}$ and $\mathcal{B}_{LB}$ are equivalent.

Claim 18 $\quad \mathcal{L}(\mathcal{A}_{NB}) = \mathcal{L}(\mathcal{B}_{LB})$

Proof: The formal proof is presented in [CY95], Proposition 4.2.2.

Informally, let $\rho : (V_0, W_0), (V_1, W_1), \ldots$ be an accepting run of $\mathcal{B}_{LB}$ over some model σ, and $0 < i_1 < i_2, \ldots$ be the positions at which ρ is accepting. Then from the construction of $\mathcal{B}_{LB}$, for each state $q' \in W_{i_{j+1}}$ there exists a state $q \in W_{i_j}$, $j \geq 0$, and a a finite run-segment of $\mathcal{A}_{NB}$ over $(\sigma, i_j), \ldots, (\sigma, i_{j+1})$ which leads from q to q' while visiting an accepting state of $\mathcal{A}_{NB}$. The fact that these segments can be concatenated into an accepting run of $\mathcal{A}_{NB}$ follows from Ramsey theorem.

The proof of the second direction ($\mathcal{L}(\mathcal{B}_{LB}) \rightarrow \mathcal{L}(\mathcal{A}_{NB})$) is based on the following Lemma (for the proof of Lemma 19 see [CY95]).

Lemma 19 *Let $\mathcal{A} = \{Q, Q_0, \delta, F\}$ be a non-deterministic Büchi automaton. A model σ is accepted by $\mathcal{A}$ if and only if σ can be written as a sequence of model-segments $\sigma = \sigma_0\sigma_1\ldots$, satisfying the following properties: there exists a state $f \in F$ and a set $R \subseteq Q$ containing f such that*

- $\sigma_0 \models \Delta(Q_0, Q')$ *for some* $Q' \subseteq Q$ *such that* $f \in Q'$.
- $\sigma_i \models \Delta(R, R) \wedge \Delta(\{f\}, R)$ *for every* $i > 0$.

Let σ be a model accepted by $\mathcal{A}_{NB}$ and $\rho^{\mathcal{A}} : q_0, q_1, \ldots$ be the accepting run. Following Lemma 19, we can decompose σ into $\sigma_0, \sigma_1, \ldots$, and find a state $f \in F$ and a set of states $R \subseteq Q$ associated with σ, satisfying Lemma 19. We construct an accepting run $\rho^{\mathcal{B}}$ of $\mathcal{B}_{LB}$ over σ as follows. The initial run-segment begins with location q_0 and ends in location $(\emptyset, \{f\})$. Since $(\emptyset, \{f\}) \in F^{\mathcal{B}}$, then $(\emptyset, \{f\})$ is a deterministic state in $Q^{\mathcal{B}}$, and the rest of the run is uniquely defined by $\delta^{\mathcal{B}}$ and σ. Let $i_0, i_1 \ldots$ be the position indices correspondig to the begininning of the sequences $\sigma_0, \sigma_1, \ldots$. Based on the construction of $\mathcal{B}_{LB}$ and Lemma 19, it can be shown that for every position index i_j, $j \geq 0$, there exists a position index k_j, $i_j \leq k_j \leq i_{j+1}$ such that $\rho^{\mathcal{B}}(k_j)$, the location at position k_j, is an accepting location of $\mathcal{B}_{LB}$.

■

Based on the automata $\mathcal{A}_{NB}$ and $\mathcal{B}_{LB}$, we define a deterministic automaton $\mathcal{C}_{DR} = \langle Q^{\mathcal{C}}, q_0^{\mathcal{C}}, \delta^{\mathcal{C}}, C^{\mathcal{C}}\rangle$ with Rabin acceptance conditions, as follows. Let $k = 2|Q^{\mathcal{B}} - Q^{\mathcal{A}}|$ and $Q^{\mathcal{C}}$ be the set of all pairs of the form $(S, (A_1, \ldots, A_k))$, such that

$$
\begin{array}{l}
S \subseteq Q^{\mathcal{A}}, \\
A_i \in (Q^{\mathcal{B}} - Q^{\mathcal{A}} \cup \{\perp\}), \text{ for every } i, 1 \leq i \leq k, \\
A_i \neq \perp \longrightarrow A_i \neq A_j \text{ for all } i \neq j.
\end{array}
$$

Without loss of generality, assume an ordering on the set $Q^{\mathcal{A}}$.

The initial location of $\mathcal{C}_{DR}$ is

$$q_0^{\mathcal{C}} = (Q_0, (\perp, \ldots, \perp)).$$

The transition assertion $\delta^{\mathcal{C}}$ is defined as follows. Let s be a state and $(S, (A_1, \ldots, A_k))$, $(S', (A'_1, \ldots, A'_k))$ be two locations in $Q^{\mathcal{C}}$, then

$$s \models \delta^{\mathcal{C}}((S, (A_1, \ldots, A_k)), (S', (A'_1, \ldots, A'_k))) \tag{6}$$

iff $s \models \Delta^{\mathcal{A}}(S, S')$ and the elements of $(A'_1, \ldots, A'_k)$ are defined from left to right, for $i = 1, \ldots, k$,

as follows.

$$A'_i = \left\{ \begin{array}{ll} X & \text{If } A_i \neq \bot, s \models \delta^{\mathcal{B}}(A_i, X) \text{ and } X \neq A'_j \text{ for all } j < i, \text{ Otherwise} \\ \bot & \text{If } A_i \neq \bot \text{ and } s \models \delta^{\mathcal{B}}(A_i, A'_j) \text{ for some } j < i, \text{ Otherwise} \\ Y & \text{If } A_i = \bot, \text{ and } l \text{ is the minimal index } l > i, \text{ such that} \\ & s \models \delta^{\mathcal{B}}(A_l, A'_j) \text{ for no } j < i, \text{and } s \models \delta^{\mathcal{B}}(A_l, Y), \text{ Otherwise} \\ (\emptyset, \{q\}) & \text{If } A_i = \bot, \text{ and } q \in S' \cap F^{\mathcal{A}} \text{ is the least element of } S' \cap F^{\mathcal{A}} \text{ such that} \\ & A'_j \neq (0, \{q\}) \text{ for all } j < i, \text{ Otherwise} \\ \bot & \end{array} \right\} \quad (7)$$

The accepting condition of $\mathcal{C}_{DR}$ is a Rabin condition, where for every i, $1 \leq i \leq k$,

$$\begin{array}{lcl} L_i & = & \{(S, (A_1, \ldots, A_k)) \mid (S, (A_1, \ldots, A_k)) \in Q^{DR} \text{ and } A_i \in F^{\mathcal{B}}\} \\ U_i & = & \{(S, (A_1, \ldots, A_k)) \mid (S, (A_1, \ldots, A_k)) \in Q^{DR} \text{ and } A_i = \bot\}\} \end{array}$$

$$acc_{\mathcal{C}_{DR}}(y) = \bigvee_{i \in [1..k]} \Box \Diamond in\_L_i(y) \wedge \Diamond \Box \neg in\_U_i(y)$$

Given a state s, a set of locations Q and a transition assertion δ over Q, we use the notation

$$\delta^{-1}(q) = \{r \mid r, q \in Q, s \models \delta(r, q)\}.$$

The following Claim states that the two automata $\mathcal{B}_{LB}$ and $\mathcal{C}_{DR}$, are equivalent.

Claim 20 $\mathcal{L}(\mathcal{B}_{LB}) = \mathcal{L}(\mathcal{C}_{DR})$

Proof: For a formal proof, see [McN66].

Informally, let σ be a model and $\rho^{\mathcal{B}} : q_0, q_1, \ldots$ be an accepting run of $\mathcal{B}_{LB}$ over σ. The model σ and the automaton $\mathcal{B}_{LB}$ define a unique run $\rho^{\mathcal{C}}$ of $\mathcal{C}_{DR}$ over σ. We have to show that $\rho^{\mathcal{C}}$ is an accepting run. The locations of $\rho^{\mathcal{C}}$ are of the form $(S, (A_1, \ldots, A_k)$. Since $\rho^{\mathcal{B}}$ is an accepting run of $\mathcal{B}_{LB}$, then for infinitely many position indices i_j, $j \geq 0$, $\rho^{\mathcal{B}}(i_j) \in F^{\mathcal{B}}$. From the construction of $\mathcal{C}_{DR}$, then for some index l_0, $1 \leq l_0 \leq k$, $\rho^{\mathcal{C}}(i_0)[A_{l_0}]$, the value of A_{l_0} at $\rho^{\mathcal{C}}(i_0)$, equals $\rho^{\mathcal{B}}(i_0)$, and therefore $\rho^{\mathcal{C}}(i_0) \in L_{l_0}$. Similarly, for some index $1 \leq l_1 \leq l_0$, $\rho^{\mathcal{C}}(i_1)[A_{l_1}] = \rho^{\mathcal{B}}(i_1)$ and thus $\rho^{\mathcal{C}}(i_1) \in L_{l_1}$. We get an infinite sequence of the form

$$\rho^{\mathcal{C}}(i_0) \in L_{l_0}, \rho^{\mathcal{C}}(i_1) \in L_{l_1}, \ldots$$

where $k \geq l_0 \geq l_1 \geq l_2 \ldots$ and for all $j \geq 0$, $l_j \geq 1$. Since the sequence $l_0, l_1, \ldots$ is a finitely descending sequence, there exists an $r \geq 0$, such that for all $j \geq r$, $l_j = l_{j+1}$. Let $l_r = m$ $(1 \leq m \leq k)$. This corresponds to the fact that from some position of $\rho^{\mathcal{C}}$, the value of A_m is never $\bot$. We conclude that for all $j \geq r$, $\rho^{\mathcal{B}}(i_j) \in L_m$ and $\rho^{\mathcal{B}}(i_j) \notin U_m$. namely, $\rho^{\mathcal{C}}$ is an accepting run of $\mathcal{B}_{DR}$.

Next, let σ be a model and $\rho^{\mathcal{C}}$ be an accepting run of $\mathcal{C}_{DR}$ over σ. We show how to construct an accepting run $\rho^{\mathcal{B}}$ of $\mathcal{B}_{LB}$ over σ. Let m, $0 \leq m \leq k$ be the accepting index of $\rho^{\mathcal{C}}$, namely $\Box \Diamond in\_L_m(\rho^{\mathcal{C}}) \wedge \Diamond \Box \neg in\_U_m(\rho^{\mathcal{C}})$. Let r be the smallest position index such that $\rho^{\mathcal{C}}(r) \notin U_m$. We construct $\rho^{\mathcal{B}}$ such that for every $i \geq r$, $\rho^{\mathcal{B}}(i) = \rho^{\mathcal{C}}(i)[A_m]$.

For every $0 \leq i < r$, if $\rho^{\mathcal{C}}(i)[A_l] = (\delta^{\mathcal{B}})^{-1}\rho^{\mathcal{B}}(i+1)$ for some l, $1 \leq l \leq k$, then $\rho^{\mathcal{B}}(i) = \rho^{\mathcal{C}}(i)[A_l]$. Otherwise, $\rho^{\mathcal{B}}(i) = (\delta^{\mathcal{B}})^{-1}\rho^{\mathcal{B}}(i+1) \cap \rho^{\mathcal{C}}(i)[S]$. It is straightforward to show that $\rho^{\mathcal{B}}$ is an accepting run of $\mathcal{B}_{LB}$.

∎

Corollary 21 $\quad \mathcal{L}(\mathcal{A}_{NB}) = \mathcal{L}(\mathcal{C}_{DR})$

Proof: An immediate result of Claims 18 and 20.

∎

5.3 Inductive Construction of Automata

In the following, we show how to construct for each formula φ an NB-automaton (non-deterministic Büchi automaton) which is congruent to φ. The construction is inductive, showing first how to construct an automaton congruent to a proposition p, and then presenting, for each operator of the logic, a construction showing how to build an automaton that corresponds to a formula using this operator from the automata corresponding to its operands.

To reduce the number of considered constructions, we replace all occurrences of the $\mathcal{U}$ operator with the temporal operators $\Box$ and $\bigcirc$, using the following congruence:

$$p \mathcal{U} q \quad \Leftrightarrow \quad \exists t: \; t \wedge \Box(t \rightarrow q \vee (p \wedge \bigcirc t)) \wedge \neg \Box t \tag{8}$$

Similarly, we replace all occurrences of the $\mathcal{S}$ operator with the temporal operators $\boxminus$ and $\ominus$, using the following congruence:

$$p \mathcal{S} q \quad \Leftrightarrow \quad \exists t: \; t \wedge \boxminus(t \rightarrow q \vee (p \wedge \ominus t)) \tag{9}$$

We proceed to present the construction of an NB-automata $\mathcal{A}_\varphi$ congruent to the formula φ by induction on the structure of φ. We use the axiomatic system **Qx** to show the provability of the congruence $\varphi \Leftrightarrow \chi_{\mathcal{A}_\varphi}$.

Case: φ is a proposition

Let p be a proposition and $\mathcal{A}_p : (Q, Q_0, \delta, F)$ be an NB-automaton given by:

$$\begin{array}{lcl} Q & = & \{q_0, q_1\} \\ Q_0 & = & \{q_0\} \\ F & = & \{q_1\} \\ \delta(q_0, q_1) & = & p \wedge t \\ \delta(q_0, q_0) & = & \delta(q_1, q_1) = \neg t \\ \delta(q_1, q_0) & = & \mathrm{F} \end{array}$$

Claim 22 *The automaton $\mathcal{A}_p$ is congruent to p. Furthermore, $\rho : r_1, r_2, \ldots$ is a j-approving run in $\mathcal{A}_p$, iff for every $i \leq j$, $r_i = q_0$ and for every $i > j$, $r_i = q_1$.*

Proof: Assume σ is j-approved by $\mathcal{A}_p$. This means that $\mathcal{A}_p$ accepts σ', the j-marked variant of σ. The accepting run of $\mathcal{A}_p$ over σ' must move from q_0 to q_1 at some step i, and this is possible only if this i satisfies

$$\sigma'_i[p] = \sigma'_i[t] = \mathrm{T}.$$

Since σ' is j-marked, then $i = j$ and we conclude that

$$\sigma'_j[p] = \sigma'_j[t] = \mathrm{T},$$

i.e. $(\sigma, j) \models p$.

Assume $(\sigma, j) \models p$, i.e., $\sigma_j[p] = \mathrm{T}$. Then σ', the j-marked variant of σ satisfies $\sigma'_j[p] = \sigma'_j[t] = \mathrm{T}$ and is accepted by a run that moves from q_0 to q_1 at step j, remaining at q_1 thereafter. This run is accepting, showing that σ' is j-approved by $\mathcal{A}_p$.

∎

Claim 23 $\quad \vdash p \Leftrightarrow \chi_{\mathcal{A}_p}$

Proof: The first direction

$$\vdash p \Rightarrow \exists t, y : S(t) \wedge app\_r_{\mathcal{A}}(y)$$

1.	$p \wedge S(t) \wedge \boxminus(y = q_0) \wedge \widehat{\Box}(y = q_1)$ $\Rightarrow S(t) \wedge app\_r_{\mathcal{A}}(y)$	Claim 22 + PTL completeness
2.	$p \wedge S(t) \wedge \exists y : \boxminus(y = q_0) \wedge \widehat{\Box}(y = q_1)$ $\Rightarrow \exists y : S(t) \wedge app\_r_{\mathcal{A}}(y)$	$\exists\exists$-**INTR** 1
3.	$\Box \exists y : \boxminus(y = q_0) \wedge \widehat{\Box}(y = q_1)$	Claim 10
4.	$p \wedge S(t) \Rightarrow \exists y : S(t) \wedge app\_r_{\mathcal{A}}(y)$	**TR** 2, 3
5.	$p \wedge \exists t : S(t) \Rightarrow \exists y, t : S(t) \wedge app\_r_{\mathcal{A}}(y)$	$\exists\exists$-**INTR** 4
6.	$\Box \exists t : S(t)$	Claim 10
7.	$p \Rightarrow \exists y, t : S(t) \wedge app\_r_{\mathcal{A}}(y)$	**TR** 5, 6

The second direction

$$\vdash S(t) \wedge app\_r_{\mathcal{A}}(y) \Rightarrow p,$$

is a direct result of claim 22 and the completeness of the axiomatic system with respect to PTL.

∎

Case: φ is of the form $p \vee q$

By induction, we assume that we have already constructed the automata congruent to p and q, given by $\mathcal{A}_p = (Q^p, Q_0^p, \delta^p, F^p)$ and $\mathcal{A}_q = (Q^q, Q_0^q, \delta^q, F^q)$ respectively. Without loss of generality, we assume that $Q^p \cap Q^q = \emptyset$. The automaton $\mathcal{A}_{p \vee q} = (Q, Q_0, \delta, F)$ is given by:

$$\begin{array}{lcl} Q & : & Q^p \cup Q^q \\ Q_0 & : & Q_0^p \cup Q_0^q \\ F & : & F^p \cup F^q \end{array}$$

For every $q_i, q_j \in Q$,

$$\delta(q_i, q_j) : \begin{cases} \delta^p(q_i, q_j) & \text{if } q_i, q_j \in Q^p \\ \delta^q(q_i, q_j) & \text{if } q_i, q_j \in Q^q \\ \text{F} & \text{Otherwise} \end{cases}$$

Claim 24 *For every model σ and position $j \geq 0$, σ is j-approved by $\mathcal{A}_{p \vee q}$ iff it is j-approved by either $\mathcal{A}_p$ or $\mathcal{A}_q$. Furthermore, $\rho : q_0, q_1, \ldots$ is a j-approving run of $\mathcal{A}_{p \vee q}$ over σ iff $\rho : q_0, q_1, \ldots$ is a j-approving run of either $\mathcal{A}_p$ or $\mathcal{A}_q$ over σ.*

Proof: Assume σ is j-approved by either $\mathcal{A} = \mathcal{A}_p$ or $\mathcal{A} = \mathcal{A}_q$. Let $\rho : q_0, q_1, \ldots$ be the j-approving run of $\mathcal{A}$ over σ'. Then obviously, ρ is a j-approving run over σ' in $\mathcal{A}_{p \vee q}$.

Assume σ' is j-approved by $\mathcal{A}_{p \vee q}$. Then, there exists a j-approving run $\rho : q_0, q_1, \ldots$ over σ' in $\mathcal{A}_{p \vee q}$, which is a j-approving run over σ' of either $\mathcal{A}_p$ or $\mathcal{A}_q$, as can be seen from the definition of δ.

∎

Claim 25 $\quad \vdash \chi_{\mathcal{A}_p} \vee \chi_{\mathcal{A}_q} \Leftrightarrow \chi_{\mathcal{A}_{p \vee q}}$

Proof: In order to prove

$$\vdash \chi_{\mathcal{A}_p} \vee \chi_{\mathcal{A}_q} \Rightarrow \chi_{\mathcal{A}_{p \vee q}},$$

it suffices to prove separately $\vdash \chi_{\mathcal{A}_p} \Rightarrow \chi_{\mathcal{A}_{p \vee q}}$ and $\vdash \chi_{\mathcal{A}_q} \Rightarrow \chi_{\mathcal{A}_{p \vee q}}$. To prove $\vdash \chi_{\mathcal{A}_p} \Rightarrow \chi_{\mathcal{A}_{p \vee q}}$, we apply rule QT to the entailment

$$S(t) \wedge app\_r_{\mathcal{A}_p}(y) \Rightarrow S(t) \wedge app\_r_{\mathcal{A}_{p \vee q}}(y)$$

The validity of the last entailment follows from Claim 24, by which, any accepting run of $\mathcal{A}_p$ is also an accepting run of $\mathcal{A}_{p \vee q}$. Provability by Qx follows from the completeness of PTL. Provability of the entailment $\chi_{\mathcal{A}_q} \Rightarrow \chi_{\mathcal{A}_{p \vee q}}$ is established similarly.

To prove the second direction

$$\vdash \chi_{\mathcal{A}_{p \vee q}} \Rightarrow \chi_{\mathcal{A}_p} \vee \chi_{\mathcal{A}_q},$$

we apply rule QT to the entailment

$$S(t) \ \wedge \ app\_r_{\mathcal{A}_{p \vee q}}(x) \Rightarrow S(t) \ \wedge \ (app\_r_{\mathcal{A}_p}(x) \vee app\_r_{\mathcal{A}_q}(x))$$

As for the first direction, the validity of this entailment follows from Claim 24, and provability follows from PTL completeness.

∎

Let $\eta(t)$ be a propositional assertion. We denote by $\eta[\mathrm{T}]$ the formula $\eta[t \leftarrow \mathrm{T}]$, i.e., the formula obtained by replacing every occurence of t by T. Similarly, we denote $\eta[t \leftarrow \mathrm{F}]$ by $\eta[\mathrm{F}]$.

Let $S \subseteq Q$ be a subset of automaton locations (possibly $S = Q$). We denote by S' the primed version of the subset S, namely

$$S' = \{q' \mid q \in S\}.$$

Case: φ is of the form $\bigcirc p$

Assume that $\mathcal{A}_p : (Q^p, Q_0^p, \delta^p, F^p)$ is an automaton congruent to p.

The automaton $\mathcal{A}_{\bigcirc p} = (Q, Q_0, \delta, F)$ is given by:

$$\begin{array}{lcl} Q & : & Q^p \cup (Q^p)' \\ Q_0 & : & Q_0^p \\ F & : & F^p \end{array}$$

For every $q_i, q_j \in Q^p$, let $\delta^p(q_i, q_j) = \eta_{ij}(t)$. Then

$$\begin{array}{lcl} \delta(q_i, q_j) & : & \neg t \ \wedge \ \eta_{ij}[\mathrm{F}] \\ \delta(q_i, q'_j) & : & t \ \wedge \ \eta_{ij}[\mathrm{F}] \\ \delta(q'_i, q_j) & : & \neg t \ \wedge \ \eta_{ij}[\mathrm{T}] \\ \delta(q'_i, q'_j) & : & \mathrm{F} \end{array}$$

Claim 26 *For every model σ and position $j \geq 0$, σ is j-approved by $\mathcal{A}_{\bigcirc p}$ iff it is $(j+1)$-approved by $\mathcal{A}_p$. Furthermore if $\widetilde{\rho} : \widetilde{q}_0, \widetilde{q}_1, \ldots$ is the j-approving run of $\mathcal{A}_{\bigcirc p}$ and $\rho : q_0, q_1, \ldots$ is the $(j+1)$-approving run of $\mathcal{A}_p$ then $\widetilde{q}_i = q'_i$ for $i = (j+1)$, and $\widetilde{q}_i = q_i$ for all $i \neq (j+1)$.*

Proof: Assume that σ is j-approved by $\mathcal{A}_{\bigcirc p}$. Let σ' be the j-marked variant of σ. Then there exists an accepting run

$$\rho : \ldots \longrightarrow q_j \overset{t \wedge \eta_{j,j+1}[\mathrm{F}]}{\longrightarrow} q'_{j+1} \overset{\neg t \wedge \eta_{j+1,j+2}[\mathrm{T}]}{\longrightarrow} q_{j+2} \longrightarrow \ldots$$

over σ' in $\mathcal{A}_{\bigcirc p}$. It is easy to see that the sequence

$$\ldots \longrightarrow q_j \overset{\eta_{j,j+1}[\mathrm{F}]}{\longrightarrow} q_{j+1} \overset{\eta_{j+1,j+2}[\mathrm{T}]}{\longrightarrow} q_{j+2} \longrightarrow \ldots$$

is an accepting run of $\mathcal{A}_p$ over σ'', the $(j+1)$-marked variant of σ.

Assume that σ is $(j+1)$-approved by $\mathcal{A}_p$. Let σ' be the $(j+1)$-marked variant of σ. Then there exists an accepting run

$$\rho : \ldots, q_j, q_{j+1}, q_{j+2} \ldots$$

over σ' in $\mathcal{A}_p$. It follows that

$$\begin{array}{ll} (\sigma', j) \models \neg t \wedge \delta^p(q_j, q_{j+1}) & \text{implying } (\sigma', j) \models \delta^p(q_j, q_{j+1})[\mathrm{F}] \\ (\sigma', j+1) \models t \wedge \delta^p(q_{j+1}, q_{j+2}) & \text{implying } (\sigma', j+1) \models \delta^p(q_{j+1}, q_{j+2})[\mathrm{T}] \end{array}$$

It follows that

$$\begin{array}{l} (\sigma'', j) \models \delta(q_j, q'_{j+1}) \\ (\sigma'', j+1) \models \delta(q'_{j+1}, q_{j+2}) \end{array}$$

where σ'' is the j-marked variant of σ (and σ'). Obviously, $\rho' : \ldots, q_j, q'_{j+1}, q_{j+2}, \ldots$ is an accepting run of $\mathcal{A}_{\bigcirc p}$ over σ''.

∎

Claim 27 $\quad \vdash \chi_{\mathcal{A}_{\bigcirc p}} \Leftrightarrow \bigcirc \chi_{\mathcal{A}_p}$

Proof: To prove the first direction

$$\vdash S(t) \wedge app\_r_{\mathcal{A}_{\bigcirc p}}(x, t) \Rightarrow \bigcirc \exists t', y : S(t') \wedge app\_r_{\mathcal{A}_p}(y, t'),$$

we apply rule QT to the entailment

$$S(t) \wedge app\_r_{\mathcal{A}_{\bigcirc p}}(x, t) \Rightarrow \bigcirc(S(\ominus t) \wedge app\_r_{\mathcal{A}_p}(unprime(x), \ominus t)) \tag{10}$$

where, for each $q \in Q^p$, $unprime(q) = unprime(q') = q$. The validity of entailment (10) follows from Claim 26. The provability of (10) follows from its validity and PTL completeness.
To prove the second direction

$$\vdash \bigcirc(S(t) \wedge app\_r_{\mathcal{A}_p}(y, t)) \Rightarrow \exists t', x : S(t') \wedge app\_r_{\mathcal{A}_{\bigcirc p}}(x, t'),$$

we apply rule QT to the entailment

$$\bigcirc(S(t) \wedge app\_r_{\mathcal{A}_p}(y, t)) \Rightarrow S(\bigcirc t) \wedge app\_r_{\bigcirc \mathcal{A}_p}(x, \bigcirc t)[\alpha], \tag{11}$$

where α is the substitution

$$\alpha : \ [x \leftarrow \ \textbf{if} \ \ t \ \textbf{then} \ \ y' \ \textbf{else} \ \ y]$$

Similar to (10), the validity of (11) follows from Claim 26, and provability follows from its validity and PTL completeness.

∎

Case: φ is of the form $\ominus p$

The automaton $\mathcal{A}_{\ominus p} = (Q, Q_0, \delta, F)$ is given by:

$$\begin{array}{lcl} Q & : & Q^p \cup (Q^p)' \\ Q_0 & : & Q_0^p \\ F & : & F^p \end{array}$$

For every $q_i, q_j \in Q^p$, let $\delta^p(q_i, q_j) = \eta_{ij}(t)$. Then

$$\begin{array}{lcl} \delta(q_i, q_j) & : & \neg t \wedge \eta_{ij}[\mathrm{F}] \\ \delta(q_i, q_j') & : & \neg t \wedge \eta_{ij}[\mathrm{T}] \\ \delta(q_i', q_j) & : & t \wedge \eta_{ij}[\mathrm{F}] \\ \delta(q_i', q_j') & : & \mathrm{F} \end{array}$$

Claim 28 *For every model σ and position $j > 0$, σ is j-approved by $\mathcal{A}_p$ iff it is $(j-1)$-approved by $\mathcal{A}_{\ominus p}$.*

Claim 29 $\quad \vdash \chi_{\mathcal{A}_{\ominus p}} \Leftrightarrow \ominus \chi_{\mathcal{A}_p}$

The proofs of claims 28 and 29 are similar to the proofs of claims 26 and 27, respectively.

Case: φ is of the form $\Diamond p$

The automaton $\mathcal{A}_{\Diamond p} = (Q, Q_0, \delta, F)$ is defined as follows:

$$\begin{array}{lcl} Q & : & Q^p \cup (Q^p)' \\ Q_0 & : & Q_0^p \\ F & : & (F^p)' \end{array}$$

For every $q_i, q_j \in Q^p$, let $\delta^p(q_i, q_j) = \eta_{ij}(t)$. Then

$$\begin{array}{lcl} \delta(q_i, q_j) & : & \eta_{ij}[\mathrm{F}] \\ \delta(q_i, q_j') & : & \eta_{ij}[\mathrm{T}] \\ \delta(q_i', q_j') & : & \neg t \wedge \eta_{ij}[\mathrm{F}] \\ \delta(q_i', q_j) & : & \mathrm{F} \end{array}$$

Claim 30 *For every model σ and position $j \geq 0$, σ is j-approved by $\mathcal{A}_{\Diamond p}$ iff it is k-approved by $\mathcal{A}_p$, for some $k \geq j$. Furthermore if $\tilde{\rho} : \tilde{q_0}, \tilde{q_1}, \ldots$ is the j-approving run of $\mathcal{A}_{\Diamond p}$ and $\rho : q_0, q_1, \ldots$ is the k-approving run of $\mathcal{A}_p$ then $\tilde{q_i} = q_i$ for all $i \leq k$ and $\tilde{q_i} = q_i'$ for all $i > k$.*

Proof: First, assume that σ is j-approved by $\mathcal{A}_{\Diamond p}$. Let σ' be the j-marked variant of σ. Then, there exists an accepting run

$$\rho : \ldots, q_j, q_{j+1}, \ldots$$

of $\mathcal{A}_{\diamond p}$ over σ', and a position $k \geq j$, such that $q_k \in Q^p$ and $q_i \in (Q^p)'$ for all $i > k$. It follows that $(\sigma', k) \models \eta_{k,k+1}[\mathrm{T}]$.

Let σ'' be the k-marked variant of σ. Then obviously, $unprime(\rho) = \ldots, q_k, unprime(q_{k+1}), \ldots$ is an accepting run of $\mathcal{A}_p$ over σ'', for some $j \leq k$.

Next, assume that σ is k-approved by $\mathcal{A}_p$ for some $k \geq j$. Let

$$\rho : \ldots, q_k, q_{k+1}, \ldots$$

be the accepting run of $\mathcal{A}_p$ over $\sigma' : s_0, s_1, \ldots$, the k-marked variant of σ. Obviously $s_i \models \eta_{i,i+1}[\mathrm{F}]$ for every $i \neq k$, and $s_k \models \eta_{k,k+1}[\mathrm{T}]$.

Then

$$\rho' : \ldots, q_k, q'_{k+1}, \ldots$$

is an accepting run of $\mathcal{A}_{\diamond p}$ over σ'', the j-marked variant of σ.

∎

Claim 31 $\quad \vdash \chi_{\mathcal{A}_{\diamond p}} \Leftrightarrow \diamond\, \chi_{\mathcal{A}_p}$

Proof: To prove the first direction

$$\vdash S(t) \ \wedge \ app\_r_{\mathcal{A}_{\diamond p}}(x,t) \Rightarrow \diamond\, \exists t', y : \ S(t') \ \wedge \ app\_r_{\mathcal{A}_p}(y,t'),$$

we apply rule QT to the entailment (provable by Claim 30 and PTL completeness)

$$S(t) \ \wedge \ app\_r_{\mathcal{A}_{\diamond p}}(x,t) \Rightarrow \diamond(S(t') \ \wedge \ app\_r_{\mathcal{A}_p}(unprime(x,t')))[\alpha],$$

where α is the substitution

$$\alpha : \ \ [t' \leftarrow \neg primed(x) \ \wedge \ primed(\bigcirc\, x)].$$

and, for a variable x whose values are in $Q^p \cup (Q^p)'$, $primed(x)$ is a boolean expression defined by

$$primed(x) : x \in (Q^p)'.$$

The proof of the second direction

$$\vdash \diamond(S(t) \ \wedge \ app\_r_{\mathcal{A}_p}(y)) \Rightarrow \exists t', x : \ S(t') \ \wedge \ app\_r_{\mathcal{A}_{\diamond p}}(x),$$

is given by the following sequence:

1. $\diamond(S(t) \ \wedge \ app\_r_{\mathcal{A}_p}(y,t)) \ \wedge \ S(t')$ Claim 30
   $\Rightarrow S(t') \ \wedge \ app\_r_{\mathcal{A}_{\diamond p}}(x,t')[\beta]$ + PTL completeness

where β is the substitution

$$\beta : \ \ [x \leftarrow \ \textbf{if} \ \ \ominus \diamondsuit\, t \ \textbf{then} \ \ y' \ \textbf{else} \ \ y].$$

2. $\diamond(S(t) \ \wedge \ app\_r_{\mathcal{A}_p}(y)) \ \wedge \ \exists t' : \ S(t')$
   $\Rightarrow \exists t' : \ S(t') \ \wedge \ app\_r_{\mathcal{A}_{\diamond p}}(x)[\beta]$ $\exists\exists$-INTR 1
3. $\Box\, \exists t' : \ S(t')$ Claim 10
4. $\diamond(S(t) \ \wedge \ app\_r_{\mathcal{A}_p}(y)) \Rightarrow \exists t' : \ S(t') \ \wedge \ app\_r_{\mathcal{A}_{\diamond p}}(x)[\beta]$ TR 2, 3
5. $\diamond(S(t) \ \wedge \ app\_r_{\mathcal{A}_p}(y)) \Rightarrow \exists t', x : \ S(t') \ \wedge \ app\_r_{\mathcal{A}_{\diamond p}}(x)$ QT, 4

∎

Case: φ is of the form $\diamondsuit p$

The automaton $\mathcal{A}_{\diamondsuit p} = (Q, Q_0, \delta, F)$ is given by:

$$\begin{array}{lcl} Q & : & Q^p \cup (Q^p)' \\ Q_0 & : & Q_0^p \\ F & : & (F^p)' \end{array}$$

For every $q_i, q_j \in Q^p$, let $\delta^p(q_i, q_j) = \eta_{ij}(t)$. Then

$$\begin{array}{lcl} \delta(q_i, q_j) & : & \neg t \wedge \eta_{ij}[\mathrm{F}] \\ \delta(q_i, q_j') & : & \eta_{ij}[\mathrm{T}] \\ \delta(q_i', q_j') & : & \eta_{ij}[\mathrm{F}] \\ \delta(q_i', q_j) & : & \mathrm{F} \end{array}$$

Claim 32 *For every model σ and position $j \geq 0$, σ is j-approved by $\mathcal{A}_{\diamondsuit p}$ if it is k-approved by $\mathcal{A}_p$, for some $k \leq j$.*

Claim 33 $\quad \vdash \chi_{\mathcal{A}_{\diamondsuit p}} \Leftrightarrow \diamondsuit \chi_{\mathcal{A}_p}$

The proofs of claims 32 and 33 are similar to the proofs of claims 30 and 31, respectively.

Case: φ is of the form $\exists v : p$

For every $q_i, q_j \in Q^p$ and $v \in \mathcal{V} - \{t\}$, let $\delta^p(q_i, q_j) = \eta_{ij}(v)$.

The automaton $\mathcal{A}_{\exists v: p} = (Q, Q_0, \delta, F)$ is defined as follows:

$$\begin{array}{lcll} Q & : & Q^p & \\ Q_0 & : & Q_0^p & \\ F & : & F^p & \\ \delta(q_i, q_j) & : & \eta_{i,j}[\mathrm{F}] \vee \eta_{i,j}[\mathrm{T}] & \text{for every } q_i, q_j \in Q \end{array}$$

Claim 34 *For every model σ and position $j \geq 0$, σ is j-approved by $\mathcal{A}_{\exists v: p}$ iff σ' is j-approved by $\mathcal{A}_p$, for some σ', a v-variant of σ. Furthermore, ρ is a j-approving run of $\mathcal{A}_{\exists v: p}$ over σ iff ρ is a j-approving run of $\mathcal{A}_p$ over σ'.*

Proof: Let $\rho : q_0, q_1, \ldots$ be the run by which $\mathcal{A}_{\exists v: p}$ j-approves the model $\sigma : s_0, s_1, \ldots$. We will construct $\sigma' : s_0', s_1', \ldots$, a v-variant of σ which is j-approved by $\mathcal{A}_p$. For every $j \geq 0$ we know that $s_j \models \eta_{j,j+1}[\mathrm{F}] \vee \eta_{j,j+1}[\mathrm{T}]$. We define $\sigma_j'[v] = \mathrm{F}$ iff $s_j \models \eta_{j,j+1}[\mathrm{F}]$. It is not difficult to see that $\sigma_j' \models \eta_{j,j+1}[v]$.

In the other direction, let $\sigma' : s_0', s_1', \ldots$ be a v-variant of $\sigma : s_0, s_1, \ldots$, such that σ' is j-approved by $\mathcal{A}_p$, using the run $\rho : q_0, q_1, \ldots$. We will show that σ is j-approved by $\mathcal{A}_{\exists v: p}$ using the same run ρ. For every $j \geq 0$, $s_j' \models \eta_{j,j+1}$ which is equivalent to $s_j' \models v \wedge \eta_{j,j+1}[\mathrm{T}] \vee \neg v \wedge \eta_{j,j+1}[\mathrm{F}]$, implying $s_j' \models \eta_{j,j+1}[\mathrm{T}] \vee \eta_{j,j+1}[\mathrm{F}]$, i.e., $s_j' \models \delta(q_i, q_j)$. Since s_j' agrees with s_j on all variables, except possibly v, and $\delta(q_i, q_j)$ does not refer to v, it follows that $s_j \models \delta(q_i, q_j)$.

∎

Claim 35 $\quad \vdash \chi_{\mathcal{A}_{\exists v:\, p}} \Leftrightarrow \exists v :\ \chi_{\mathcal{A}_p}$

Proof: To prove the first direction

$$\vdash S(t) \ \wedge \ app\_r_{\mathcal{A}_{\exists v:\, p}}(x) \Rightarrow \exists t', y, v :\ S(t') \ \wedge \ app\_r_{\mathcal{A}_p}(y, v),$$

we apply rule QT to the entailment

$$\vdash S(t) \ \wedge \ app\_r_{\mathcal{A}_{\exists v:\, p}}(x) \Rightarrow S(t) \ \wedge \ app\_r_{\mathcal{A}_p}(x, least\_sat(v)), \tag{12}$$

where $least\_sat(v)$ is given by the expression

$$\textbf{if} \ \ \delta^p(x, \bigcirc x)[\mathrm{F}] \ \textbf{then} \ \ \mathrm{F} \ \ \textbf{else} \ \ \mathrm{T},$$

and $\delta^p(x, \bigcirc x)[\mathrm{F}]$[3] is the formula obtained by replacing every occurence of v in the propositional assertion $\delta^p(x, \bigcirc x)$ by F.

To prove the second direction

$$\vdash \exists v :\ S(t) \ \wedge \ app\_r_{\mathcal{A}_p}(y) \Rightarrow \exists t', x :\ S(t') \ \wedge \ app\_r_{\mathcal{A}_{\exists v:\, p}}(x)$$

we apply rule QT to the entailment

$$\vdash S(t) \ \wedge \ app\_r_{\mathcal{A}_p}(y) \Rightarrow S(t) \ \wedge \ app\_r_{\mathcal{A}_{\exists v:\, p}}(y). \tag{13}$$

The validity of both (12) and (13) result from Claim 34. Provability results from their validity and PTL completeness.

∎

5.4 The Case of Negation

Assume that φ is of the form $\neg p$ and that we have already constructed the automaton $\mathcal{A}_p$ congruent to p. As discussed in subsection 5.2, we construct the sequence

$$\mathcal{A}_p \ \xrightarrow{step1} \mathcal{C}_{DR} \ \xrightarrow{step2} \mathcal{D}_{DS} \ \xrightarrow{step3} \mathcal{A}_{\neg p},$$

where $\mathcal{C}_{DR}$ is a deterministic Rabin automaton equivalent to $\mathcal{A}_p$, $\mathcal{D}_{DS}$ is a deterministic Streett automaton which is the complement of $\mathcal{C}_{DR}$, and $\mathcal{A}_{\neg p}$ is a non-deterministic Büchi automaton equivalent to $\mathcal{D}_{DS}$.

[3]Note that $\delta^p(x, \bigcirc x)[\mathrm{F}]$ is a shorthand notation for the formula $\bigvee_{i,j} at\_q_i(x) \wedge \bigcirc at\_q_j(x) \wedge \delta^p(q_i, q_j)[\mathrm{F}]$.

Negation - Step 1: $\mathcal{A}_p \longrightarrow \mathcal{C}_{DR}$. Let p be a QPTL formula and $\mathcal{A}_p = (Q, Q_0, \delta, F)$ be an NB-automaton congruent to p. Let $\mathcal{C}_{DR} = (Q^{DR}, q_0^{DR}, \delta^{DR}, C^{DR})$ be a DR-automaton equivalent to $\mathcal{A}_p$, as defined in subsection 5.2.

Claim 36 $\quad \vdash \exists x : \ acc\_r_{\mathcal{C}_{DR}}(x) \leftrightarrow \exists y : \ acc\_r_{\mathcal{A}_p}(y)$

Claim 36 is established by a sequence of claims. We first set out to prove the first implication

$$\vdash acc\_r_{\mathcal{C}_{DR}}(x) \rightarrow \exists y : \ acc\_r_{\mathcal{A}_p}(y). \tag{14}$$

Let $\sigma : s_0, s_1, \ldots$ be a model, and x be a variable whose interpretation identifies an accepting run of $\mathcal{C}_{DR}$ over σ. Since x represents an accepting run of $\mathcal{C}_{DR}$ over σ, then $\sigma \models \bigvee_{i \in [1..k]} \Box \Diamond in\_L_i \wedge \Diamond \Box \neg in\_U_i$. Let m be the minimal index, $1 \leq m \leq k$, such that $\sigma \models \Box \Diamond in\_L_m \wedge \Diamond \Box \neg in\_U_m$.

Recall that a location in Q^{DR} is of the form $(S, (A_1, \ldots, A_k))$, where $A_i = (V_i, W_i)$ (or $A_i = \bot$) and $S, V_i, W_i \subseteq Q$, for every i, $1 \leq i \leq k$. In order to identify a unique run y of $\mathcal{A}_p$ over σ, we first define a total ordering on Q such that for every non-empty $S \subseteq Q$, the function $least(S)$ specifies a unique location in S.
We can now define a prophecy scheme $y = f_m(\bigcirc y)$, as follows:

$$f_m(\bigcirc y) \ : \ \begin{cases} least(V_m \cup W_m) & \text{If } \Box(A_m \neq \bot) \text{ and } \bigcirc y \notin \bigcirc V_m \cup \bigcirc W_m \\ least(\delta^{-1}(\bigcirc y)) \cap W_m) & \text{Else, if } \Box(A_m \neq \bot) \text{ and } \delta^{-1}(\bigcirc y) \cap W_m \neq \emptyset \\ least(\delta^{-1}(\bigcirc y) \cap V_m) & \text{Else, if } \Box(A_m \neq \bot) \text{ and } \delta^{-1}(\bigcirc y) \cap V_m \neq \emptyset \\ least(\delta^{-1}(\bigcirc y) \cap S) & \text{Else, if } \delta^{-1}(\bigcirc y) \cap S \neq \emptyset \\ least(Q) & \text{Otherwise} \end{cases}$$

The purpose of function $f_m(\bigcirc y)$ is to define a descending inductive scheme by which, given the value of y at position $i+1$, we determine the value of y at position i. Note that the fuction f_m depends on the choice of m. Denote the value of y at positions i and $i+1$ by $q_i, q_{i+1} \in Q$. The definition of f states the following:

- If we are at a position i beyond which $\Box(A_m \neq \bot)$, and $q_{i+1} \notin \bigcirc V_m \cup \bigcirc W_m$, we take q_i to be the least member of $V_m \cup W_m$
- Otherwise, if we are at a position beyond which $\Box(A_m \neq \bot)$, and there exists a location $q \in W_m$ such that $s_i \models \delta(q, q_{i+1})$, we take q_i to be the least member of $\delta^{-1}(q_{i+1}) \cap W_m$.
- Otherwise, if we are at a position beyond which $\Box(A_m \neq \bot)$, and there exists a location $q \in V_m$ such that $s_i \models \delta(q, q_{i+1})$, we take q_i to be the least member of $\delta^{-1}(q_{i+1}) \cap V_m$.
- Otherwise, if $\delta^{-1}(q_{i+1}) \cap S$ is non-empty, we take q_i to be the least member of $\delta^{-1}(q_{i+1}) \cap S$.
- Otherwise, we take q_i to be the least member of Q.

Claim 37 $\quad \vdash acc\_r_{\mathcal{C}_{DR}}(x) \wedge (\Box \Diamond in\_L_m(x) \wedge \Diamond \Box \neg in\_U_m(x)) \wedge \Box(y = f_m(\bigcirc y)) \rightarrow acc\_r_{\mathcal{A}_p}(y)$

Proof: We show that the claimed implication is (semantically) valid.

Let $\sigma : s_0, s_1, \ldots$ be a model, and x be a variable whose interpretation ρ identifies an accepting run of $\mathcal{C}_{DR}$ over σ. Let m be the minimal accepting index of ρ. From the construction of $\mathcal{C}_{DR}$, we can deduce the following properties of ρ. Let l be a position beyond which $\Box(A_m \neq \bot)$, and let $i > l$. Then,

P1. $V_m(i) \cup W_m(i) \neq \emptyset$ and every location $q \in V_m(i+1) \cup W_m(i+1)$ has a predecessor in $V_m(i) \cup W_m(i)$.

P2. For every location $q \in W_m(i)$, either $q \in F$ or q has a predecessor in $W_m(i-1)$

P3. If $\rho(i)$, the location at position i, is an accepting location, then $V_m(i) = \emptyset$ and $W_m(i+1) \subseteq F$.

First, we show that every y satisfying $\Box(y = f(\bigcirc y))$ is a run of $\mathcal{A}_p$ over the considered model σ. In terms of position indices, $\Box(y = f(\bigcirc y))$ implies $y(i) = f(y(i+1))$ for all $i \geq 0$. The definition of f consists of 5 clauses. Clauses $2-4$ guarantee that $y(i)$ is a predecessor of $y(i+1)$ or, equivalently, that $y(i+1)$ is a δ-successor of $y(i)$. We claim that, if $y(i) = f(y(i+1))$ holds for all $i \geq 0$, then clauses 1 and 5 are never used.

Let l denote the position beyond which A_m is never $\bot$, namely $A_m(i) \neq \bot$ for all $i \geq l$. Clause 1 can only be used at some position $i \geq l$. To be used at i, it is necessary that $y(i+1) \notin V_m(i+1) \cup W_m(i+1)$. Consider location $y(i+2)$. If $y(i+2) \notin (V_m(i+2) \cup W_m(i+2))$ then, in the computation of $y(i+1) = f(y(i+2))$, we use clause 1 and pick $y(i+1)$ to be the least location in $V_m(i+1) \cup W_m(i+1)$, which, by property $P1$, is non-empty. Alternately, if $y(i+2) \in V_m(i+2) \cup W_m(i+2)$, then by property $P1$, either $\delta^{-1}(\bigcirc y) \cap V_m(i+1) \neq \emptyset$ or $\delta^{-1}(\bigcirc y) \cap W_m(i+1) \neq \emptyset$, and we use clauses 2 or 3 to pick $y(i+1)$ to be the least location in either $W_m(i+1)$ or $V_m(i+1)$ respectively. It follows that $y(i+1) \in W_m(i+1) \cup V_m(i+1)$ in both cases, which shows that clause 1 could not have been used for computing $y(i) = f(y(i+1))$.

By the previous argument $y(i) \in V_m(i) \cup W_m(i)$ and is a predecessor of $y(i+1)$ for all $i \geq l$. By the construction of $\mathcal{C}_{DR}$, $V_m(i) \cup W_m(i) \subseteq S(i)$ for all $i \geq 0$, and every location $q \in S(i)$ has a predecessor in $S(i-1)$ for all $i > 0$. Thus, in the evaluation of $y(i) = f(y(i+1))$ for $i < l$, we use clause 4, which shows that clause 5 is never used.

Next we show that $y = y(0), y(1), \ldots$ is an accepting run of $\mathcal{A}_p$ over σ. Let i, j, $l \leq i < j$ be two positions such that $x(i)$ and $x(j)$ are accepting locations in $\mathcal{C}_{DR}$. Let $R = \{r \mid i < r \leq j, y(r) \in W(r)\}$. Since $x(j)$ is an accepting locations in $\mathcal{C}_{DR}$, then $V_m(j) = \emptyset$, and by property $P1$, $W_m(j) \neq \emptyset$. Then, by the definition of f, $y(j) \in W_m(j)$. Thus, $R \neq \emptyset$. Let $r \in R$ be the minimal element in R. If $r = i+1$ then $y(r) \in W_m(i+1)$, and since by property $P3$, $W_m(i+1) \subseteq F$, then $y(r) \in F$. Else, if $r > i+1$ then $y(r) \in W_m(r)$ and $y(r-1) \notin W_m(r-1)$ (since r is minimal in R). By the definition of f for $y(i) \notin W_m(i)$, $y(r)$ can not have a predecessor in $W_m(r-1)$. Thus, by property $P2$, $y(r) \in F$.

Thus, for every positions i, j, $i < j$, such that $x(i)$ and $x(j)$ are accepting locations in $\mathcal{C}_{DR}$, there exists a location r, $i < r \leq j$, such that $y(r)$ is an accepting location in $\mathcal{A}_p$. Since there are infinitely many positions $i_1 < i_2 < \ldots$ at which x is accepting, y passes infinitely many times through accepting locations.

Once the validity of the implication is established, its provability follows from PTL completeness. ∎

We can now proceed with the proof of (14):

1.	$acc\_r_{\mathcal{C}_{DR}}(x) \rightarrow \bigvee_{1 \le i \le k} \underbrace{\Box \Diamond in\_L_i(x) \wedge \Diamond \Box \neg in\_U_i(x)}_{p_i(x)}$	Def. of $acc\_r_{\mathcal{C}_{DR}}$
2.	$acc\_r_{\mathcal{C}_{DR}}(x) \wedge \bigvee_{1 \le i \le k} (p_i(x) \wedge \Box(y = f_i(\bigcirc y))) \rightarrow acc\_r_{\mathcal{A}_p}(y)$	Claim 37 + PTL completeness
3.	$acc\_r_{\mathcal{C}_{DR}}(x) \wedge \bigvee_{1 \le i \le k} (p_i(x) \wedge \exists y : \Box(y = f_i(\bigcirc y))) \rightarrow \exists y : acc\_r_{\mathcal{A}_p}(y)$	$\exists\exists$-INTR 2
4.	$\bigwedge_{1 \le i \le k} \exists y : \Box(y = f_i(\bigcirc y))$	Theorem 3
5.	$acc\_r_{\mathcal{C}_{DR}}(x) \rightarrow \exists y : acc\_r_{\mathcal{A}_p}(y)$	TR 3, 1, 4

∎

Next, we prove the second implication

$$\vdash acc\_r_{\mathcal{A}_p}(y) \rightarrow \exists x : acc\_r_{\mathcal{C}_{DR}}(x), \tag{15}$$

Let $\sigma : s_0, s_1, \ldots$ be a model, and y be a variable whose interpretation identifies an accepting run of $\mathcal{A}_p$ over σ. In order to identify a unique run x of $\mathcal{C}_{DR}$ over σ, we define a history scheme $x = h(\ominus x)$, where $h(\ominus x)$ is defined as follows:

$$h(\ominus x) : \begin{cases} q_0^{DR} & \text{If } \textit{first} \\ min\ q\ .\ \delta^{DR}(\ominus x, q) & \text{Otherwise} \end{cases}$$

The purpose of function $h(\ominus x)$ is to define a ascending inductive scheme by which, given the value of x at position $i-1$, we determine the value of x at position i.

The definition of $h(\ominus x)$ relies on the determinization to construct the unique run of $\mathcal{C}_{DR}$ over the considered model $\sigma : s_0, s_1, \ldots$. It sets x_0 to q_0^{DR}, the single initial location of $\mathcal{C}_{DR}$. Then, for every $i \ge 1$, $x(i)$ is defined to be the unique $\mathcal{C}_{DR}$ location $q \in Q^{DR}$ satisfying $s_{i-1} \models \delta^{DR}(x(i-1), q)$. Note that the historic function $h(\ominus x)$ does not refer to y.

Claim 38 $\quad \vdash acc\_r_{\mathcal{A}_p}(y) \wedge \Box(x = h(\ominus x)) \rightarrow acc\_r_{\mathcal{C}_{DR}}(x)$

Proof: A direct result of Corollary 21 and PTL completeness.

∎

We can now proceed with the proof of (15):

1.	$acc\_r_{\mathcal{A}_p}(y) \wedge \Box(x = h(\ominus x)) \rightarrow acc\_r_{\mathcal{C}_{DR}}(x)$	Claim 38
2.	$acc\_r_{\mathcal{A}_p}(y) \wedge \exists x : \Box(x = h(\ominus x)) \rightarrow \exists x : acc\_r_{\mathcal{C}_{DR}}(x)$	$\exists\exists$-INTR 1
3.	$\exists x : \Box(x = h(\ominus x))$	Theorem 1
4.	$acc\_r_{\mathcal{A}_p}(y) \rightarrow \exists x : acc\_r_{\mathcal{C}_{DR}}(x)$	TR 2, 3

∎

Negation - Step 2: $\mathcal{C}_{DR} \longrightarrow \mathcal{D}_{DS}$.

Let $\mathcal{C}_{DR} = (Q^{DR}, q^{DR}, \delta^{DR}, C^{DR})$ be a **DR**-automaton as defined in step 1. We define a **DS**-automaton $\mathcal{D}_{DS} = (Q^{DS}, q_0^{DS}, \delta^{DS}, C^{DS})$ as follows:

- $Q^{DS} = Q^{DR}, q_0^{DS} = q_0^{DR}, \delta^{DS} = \delta^{DR}$
- C^{DS} - For every $i \in [1..k]$, L_i and U_i are defined as in $\mathcal{C}_{DR}$.
  $acc_{\mathcal{D}_{DS}}(y) : \bigwedge_{i \in [1..k]} \Diamond \Box \neg in\_L_i(y) \vee \Box \Diamond in\_U_i(y)$

Claim 39 $\quad \vdash \exists x : \ acc\_r_{\mathcal{C}_{DR}}(x) \leftrightarrow \neg \exists y : \ acc\_r_{\mathcal{D}_{DS}}(y)$

Proof: The first implication is proven as follows:

1.	$acc_{\mathcal{C}_{DR}}(x) \ \wedge \ \Box(y = x) \rightarrow \neg acc_{\mathcal{D}_{DS}}(y)$	Def. of $acc_{\mathcal{C}_{DR}}, acc_{\mathcal{D}_{DS}}$ and PTL completenes
2.	$init_{\mathcal{C}_{DR}}(x) \ \wedge \ run_{\mathcal{C}_{DR}}(x) \ \wedge \ \Box(y = x)$ $\rightarrow \underbrace{init_{\mathcal{D}_{DS}}(y) \ \wedge \ run_{\mathcal{D}_{DS}}(y)}_{p_{\mathcal{D}_{DS}}(y)}$	Def. of $\mathcal{C}_{DR}, \mathcal{D}_{DS}$ and PTL completeness
3.	$init_{\mathcal{C}_{DR}}(x) \ \wedge \ run_{\mathcal{C}_{DR}}(x) \ \wedge \ \neg \Box(y = x)$ $\rightarrow \neg p_{\mathcal{D}_{DS}}(y)$	$\mathcal{D}_{DS}$ deterministic, 2, and PTL completeness
4.	$acc\_r_{\mathcal{C}_{DR}}(x)$ $\rightarrow \neg \Box(y = x) \ \wedge \ \neg p_{\mathcal{D}_{DS}}(y) \vee \Box(y = x) \ \wedge \ \neg acc_{\mathcal{D}_{DS}}(y)$	**TR** $1, 3$
5.	$acc\_r_{\mathcal{C}_{DR}}(x) \rightarrow \neg acc\_r_{\mathcal{D}_{DS}}(y)$	**TR** 4
6.	$acc\_r_{\mathcal{C}_{DR}}(x) \rightarrow \forall y : \ \neg acc\_r_{\mathcal{D}_{DS}}(y)$	$\forall$-**INTR** 5
7.	$acc\_r_{\mathcal{C}_{DR}}(x) \rightarrow \neg \exists y : \ acc\_r_{\mathcal{D}_{DS}}(y)$	**QR** 6

The second implication is proven similarly.

∎

Negation - Step 3: $\mathcal{D}_{DS} \longrightarrow \mathcal{A}_{\neg p}$.

The following construction is taken from [Saf88] where it is attributed to Vardi. Let $\mathcal{D}_{DS} = (Q^{DS}, q_0^{DS}, \delta^{DS}, C^{DS})$ be a **DS**-automaton with k accepting pairs. We construct an equivalent **NB**-automaton $\mathcal{A}_{\neg p} = (Q, Q_0, \delta, F)$ as follows:

- Q : $\{(q, S_1, S_2) \mid q \in Q^{DS}, S_i \subseteq \{1, \ldots, k\}$ or $S_i = \{\#\}\}$ for $i = 1, 2$
- Q_0 : $\{(q_0^{DS}, \{\#\}, \{\#\})\}$
- δ : Consider $(q, S_1, S_2) \in Q$ and $q' \in Q^{DS}$. Define

  Δ_1 : $S_1 \cup \{i \mid q' \in L_i\}$
  Δ_2 : $S_2 \cup \{i \mid q' \in U_i\}$

  We say that (q', S_1', S_2') is a *syntactic successor* of (q, S_1, S_2) if one of the following holds:

- $S_1 = S_2 = \{\#\}$ and $(S'_1 = S'_2 = \{\#\}$ or $S'_1 = S'_2 = \emptyset)$.
- $\Delta_1 \subseteq \Delta_2$ and $S'_1 = S'_2 = \emptyset$.
- $\Delta_1 \not\subseteq \Delta_2$ and $S'_1 = \Delta_1$, $S'_2 = \Delta_2$.

For every (q', S'_1, S'_2) a syntactic successor of (q, S_1, S_2), we set

$$\delta((q, S_1, S_2), (q', S'_1, S'_2)) = \delta^{DS}(q, q')$$

If (q', S'_1, S'_2) is not a syntactic successor of (q, S_1, S_2), we set

$$\delta((q, S_1, S_2), (q', S'_1, S'_2)) = \text{F}$$

- $F \quad : \quad \{(q, \emptyset, \emptyset) \mid q \in Q^{DS}\}$

Claim 40 $\quad \vdash \exists x : \ acc\_r_{\mathcal{D}_{DS}}(x) \leftrightarrow \exists y : \ acc\_r_{\mathcal{A}_{\neg p}}(y)$

Proof: The proof of the first implication

$$\vdash acc\_r_{\mathcal{D}_{DS}}(x) \rightarrow \exists y : \ acc\_r_{\mathcal{A}_{\neg p}}(y), \tag{16}$$

is performed in several steps. First we observe the following claim:

Claim 41 *Let σ be a model and x be a variable whose interpretation identifies an accepting run of $\mathcal{D}_{DS}$ over σ. Then for some $j \geq 0$,*

$$(\sigma, j) \models \underbrace{\bigwedge_{i \in [1..k]} in\_L_i(x) \Rightarrow \Diamond\, in\_U_i(x)}_{stable}$$

Proof: The validity of the claim is obvious from the definition of Streett acceptance conditions. ∎

Let σ be a model and ρ be a run of $\mathcal{D}_{DS}$ over σ. In order to identify a unique run of $\mathcal{A}_{\neg p}$ over σ, we define a history scheme $y = h(\ominus y)$ as follows:

$$h(\ominus y) : \begin{cases} (q_0^{DS}, \{\#\}, \{\#\}) & \text{If } \textit{first} \\ (q', \{\#\}, \{\#\}), \text{ where } q' = \textit{min}\ q\ .\ \delta^{DS}(\ominus x, q) & \text{Else, if } \neg \textit{stable} \\ (q', \emptyset, \emptyset), \text{ where } q' = \textit{min}\ q\ .\ \delta^{DS}(\ominus x, q) & \text{Else, if } \textit{stable} \wedge \ominus \neg \textit{stable} \\ \textit{min}\ u\ .\ \delta(\ominus y, u) & \text{Otherwise} \end{cases}$$

The definition of $h(\ominus y)$ sets $y(0)$ to $(q_0^{DS}, \{\#\}, \{\#\})$, the single initial location of $\mathcal{A}_{\neg p}$. Relying on the knowledge of position j at which the run of $\mathcal{D}_{DS}$ over σ becomes stable, $y(i)$ is set to $(x(i), \{\#\}, \{\#\})$, at all non-stable positions $0, \ldots, j-1$. At position j, the value of $y(j)$ is set to $(x(j), \emptyset, \emptyset)$. Finally, at all positions $k > j$, the value of $y(k)$ is evaluated relying on δ being deterministic at all stable positions.

Claim 42 $\quad \vdash acc\_r_{\mathcal{D}_{DS}}(x) \wedge \Diamond\, stable(x) \wedge \Box(y = h(\ominus\, y)) \rightarrow acc\_r_{\mathcal{A}_{\neg p}}(y)$

Proof: To prove the validity of the claim, we observe that every y satisfying $\Box(y = h(\ominus\, y))$ is a run of $\mathcal{A}_{\neg p}$ over σ. To see that the run $y(0), y(1), \ldots$ is an accepting run, observe that S_1 and S_2 are set to $\emptyset$ on the first stable position, and at every position satisfying $S_1 \subseteq S_2$. Since x is an accepting run of $\mathcal{D}_{DS}$, then $\Box \Diamond(\Delta_1 \subseteq \Delta_2)$, which implies $\Box \Diamond(y = \bigvee_{q \in Q^{DS}}(q, \emptyset, \emptyset))$, namely, at infinitely many positions $i > j$, $y(i) \in F$.

Provability follows from the validity and completeness of the axiomatic system $\mathbf{Q_X}$ for PTL.

▟

We can now prove (16), as follows:

1.	$acc\_r_{\mathcal{D}_{DS}}(x) \wedge \Diamond\, stable(x) \wedge \Box(y = h(\ominus\, y)) \rightarrow acc\_r_{\mathcal{A}_{\neg p}}(y)$	Claim 42
2.	$acc\_r_{\mathcal{D}_{DS}}(x) \wedge \Diamond\, stable(x) \wedge \exists y : \Box(y = h(\ominus\, y)) \rightarrow \exists y : acc\_r_{\mathcal{A}_{\neg p}}(y)$	$\exists\exists$-INTR 1
3.	$\exists y : \Box(y = h(\ominus\, y))$	Theorem 1
4.	$acc\_r_{\mathcal{D}_{DS}}(x) \rightarrow \exists y : acc\_r_{\mathcal{A}_{\neg p}}(y)$	TR 2, 3, Claim 41

▟

To prove the second implication

$$\vdash acc\_r_{\mathcal{A}_{\neg p}}(y) \rightarrow \exists x : acc\_r_{\mathcal{D}_{DS}}(x),$$

we apply rule QT to the following implication

$$acc\_r_{\mathcal{A}_{\neg p}}(y) \rightarrow acc\_r_{\mathcal{D}_{DS}}(location(y)),$$

where, for $y = (q, S_1, S_2)$, $location(y) = q$. The validity of this implication is a straightforward result of the definition of $\mathcal{A}_{\neg p}$, and provability follows from the validity and the completeness of PTL.

▮

Let p be a formula and $\mathcal{A}_p$ be an NB-automaton congruent to p. Let $\mathcal{A}_{\neg p}$ be the automaton resulting from the three-step determinization construction.

Claim 43 $\quad \vdash \chi_{\mathcal{A}_{\neg p}} \Leftrightarrow \neg \chi_{\mathcal{A}_p}$

Proof:

1.	$\exists x : acc\_r_{\mathcal{A}_{\neg p}}(x) \leftrightarrow \neg \exists y : acc\_r_{\mathcal{A}_p}(y)$	Claim 36, Claims 39, 40
2.	$\Box \Diamond\!\!\!-\, first$	TR
3.	$\Box \Diamond\!\!\!-(first \wedge (\exists x : acc\_r_{\mathcal{A}_{\neg p}}(x) \leftrightarrow \neg \exists y : acc\_r_{\mathcal{A}_p}(y)))$	TR 1, 2
4.	$\Diamond\!\!\!-(first \wedge \exists x : acc\_r_{\mathcal{A}_{\neg p}}(x)) \Leftrightarrow \Diamond\!\!\!-(first \wedge \neg \exists y : acc\_r_{\mathcal{A}_p}(y))$	TR 3
5.	$\exists x : \Diamond\!\!\!-(first \wedge acc\_r_{\mathcal{A}_{\neg p}}(x)) \Leftrightarrow \neg \exists y : \Diamond\!\!\!-(first \wedge acc\_r_{\mathcal{A}_p}(y))$	QR 2, 4
6.	$\exists x : app\_r_{\mathcal{A}_{\neg p}}(x) \Leftrightarrow \neg \exists y : app\_r_{\mathcal{A}_p}(y)$	5, def. of $app\_r_{\mathcal{A}}$
7.	$\exists t : S(t) \wedge \exists x : app\_r_{\mathcal{A}_{\neg p}}(x) \Leftrightarrow \exists t : S(t) \wedge \neg \exists y : app\_r_{\mathcal{A}_p}(y)$	TR 6, $\exists\exists$-INTR
8.	$\exists t, x : S(t) \wedge app\_r_{\mathcal{A}_{\neg p}}(x) \Leftrightarrow \neg \exists t, y : S(t) \wedge app\_r_{\mathcal{A}_p}(y)$	QR 7
9.	$\chi_{\mathcal{A}_{\neg p}} \Leftrightarrow \neg \chi_{\mathcal{A}_p}$	8, Def. of χ

∎

Theorem 4 *For every* QPTL *formula* φ*, there exists a Büchi automaton* $\mathcal{A}_\varphi$ *such that*

$$\vdash \varphi \Leftrightarrow \chi_{\mathcal{A}_\varphi}.$$

Proof: A direct result of all claims in Subsections 5.3 and 5.4.

5.5 Winding it All Up

Assume that we have obtained our final automaton $\mathcal{A}_\varphi$ which is congruent (and has been formally shown to be so) to the formula φ. The last step is to test whether $\mathcal{A}_\varphi$ is initially void and show that if it is initially void then $\neg\varphi$ which is congruent to $\neg\chi_{\mathcal{A}_\varphi}$ is provable.

Claim 44 *If automaton* $\mathcal{A}_\varphi$ *is initially void, then* $\neg\chi_{\mathcal{A}}$ *is provable.*

Proof: From the definition of $\chi_{\mathcal{A}}$,

$$\neg\chi_{\mathcal{A}}: \quad \forall y,t \ : \neg S(t) \vee \boxminus(\neg \mathit{first}(y) \vee \neg \mathit{init}_{\mathcal{A}}(y) \vee \neg \mathit{run}_{\mathcal{A}}(y) \vee \Diamond \Box \neg \mathit{in\_F}^{\mathcal{A}}(y))$$

The above formula is valid iff the following formula is valid

$$\neg S(t) \vee \boxminus(\neg \mathit{first}(y) \vee \neg \mathit{init}_{\mathcal{A}}(y) \vee \neg \mathit{run}_{\mathcal{A}}(y) \vee \Diamond \Box \neg \mathit{in\_F}^{\mathcal{A}}(y)) \tag{17}$$

Finally, the validity of (17) is provable by the completeness of the PTL axiomatic system.

∎

This leads to the final result:

Theorem 5 *If the formula* φ *is valid, then it is provable in the axiomatic system* **Qx**.

Proof: Let φ be a valid formula and $\mathcal{A}_{\neg\varphi}$ be an NB-automata congruent to $\neg\varphi$. Since φ is valid, then $\neg\varphi$ is not satisfiable and $\mathcal{A}_{\neg\varphi}$ is initially void.

1. $\vdash \neg\chi_{\mathcal{A}_{\neg\varphi}}$ — $\mathcal{A}_{\neg\varphi}$ initially void , Claim 44
2. $\vdash \neg\varphi \Leftrightarrow \chi_{\mathcal{A}_{\neg\varphi}}$ — Theorem 4
3. $\vdash \neg(\neg\varphi)$ — 1, 2
4. $\vdash \varphi$ — 3

∎

6 Discussion

The general technique proposed here, proving completeness by a formal trace of an automata reduction, seems to be widely applicable. We are currently studying other versions of temporal logics, examining whether their completeness can be determined by very similar means. Among the logics that should be considered are full QPTL but with the floating notion of validity. Another interesting logic would be the future fragment of PTL, and of course, stuttering-free logics such as TLA and other variants.

Acknowledgment: We thank the referees for their helpful comments.

Appendix A

In the following we present the proofs of all lemmas, claims and theorems of section 4.

Lemma 2 *Let $\varphi(y)$ be a* QPTL *formula. Then*

$$\vdash \varphi(y) \Rightarrow \exists z : (\varphi(z) \wedge z = min\ u\,.\,\varphi(u))$$

Proof: First, define

$$\chi_i = \exists u : \varphi(u) \wedge u = k_i$$

Then

$$\min u\,.\,\varphi(u) = \textbf{ if } \chi_1 \textbf{ then } k_1 \textbf{ else-if } \ldots \textbf{ else-if } \chi_n \textbf{ then } k_n \textbf{ else } k_1$$

Note: For every $i = 1, \ldots, n$,

$$\vdash \chi_i \wedge \bigwedge_{j<i} \neg\chi_j \Rightarrow \min u\,.\,\varphi(u) = k_i \tag{18}$$

We can now proceed with the proof:

1.	$\varphi(y) \Rightarrow \bigvee_i (\varphi(y) \wedge y = k_i)$	TAU
2.	$\varphi(y) \Rightarrow \bigvee_i \chi_i$	**QT** 1
3.	$\bigvee_i \chi_i \Rightarrow \bigvee_i \underbrace{\chi_i \wedge \bigwedge_{j<i} \neg\chi_j}_{r_i}$	TAU
4.	$r_i \Rightarrow \exists z : \varphi(z) \wedge z = k_i$	Defs. of χ_i, r_i
5.	$r_i \Rightarrow \min u\,.\,\varphi(u) = k_i$	(18)
6.	$r_i \Rightarrow \exists z : \varphi(z) \wedge z = \min u\,.\,\varphi(u)$	4, 5
7.	$\bigvee_i r_i \Rightarrow \exists z : \varphi(z) \wedge z = \min u\,.\,\varphi(u)$	**TR** 6
8.	$\varphi(y) \Rightarrow \exists z : \varphi(z) \wedge z = \min u\,.\,\varphi(u)$	**TR** 2, 3, 7

∎

Lemma 3 *Let $\varphi(y)$ be an extensible formula. Then*

$$\vdash \Box\, \exists y : \varphi(y)$$

Proof: Using the induction theorem **NFX**5, we need to show that

$$\vdash \circledcirc\, \exists y : \varphi(y) \Rightarrow \exists y : \varphi(y)$$

which follows immediately from the definition of extensibility.

∎

Claim 4 *Let $\varphi(y)$ be a uniquely extensible formula. Then*

$$\vdash \varphi(y) \Rightarrow (y = \mathit{min}\ u\,.\,\varphi(u))$$

Proof:

1.	$\varphi(y) \Rightarrow \exists z :\ \varphi(z)\ \wedge\ z = \min u\,.\,\varphi(u)$	Lemma 2
2.	$\varphi(y)\ \wedge\ \varphi(z) \Rightarrow (y = z)$	unique extensibility
3.	$\varphi(y) \Rightarrow (y = \min u\,.\,\varphi(u))$	**PR** $1, 2$ and TAU

∎

Claim 5 *Let $\varphi(y)$ be uniquely extensible. Then*

$$\vdash \varphi(y) \Rightarrow \varphi(\mathit{min}\ u\,.\,\varphi(u))$$

Proof:

1.	$\varphi(y) \Rightarrow (y = \min u\,.\,\varphi(u))$	Claim 4
2.	$\boxminus\, \varphi(y) \Rightarrow\ \boxminus (y = \min u\,.\,\varphi(u))$	**TR** 1
3.	$\varphi(y) \Rightarrow\ \boxminus\, \varphi(y)$	unique extensibility
4.	$\boxminus (y = \min u\,.\,\varphi(u)) \Rightarrow \varphi(y) \leftrightarrow \varphi(\min u\,.\,\varphi(u))$	past-dependence
5.	$\varphi(y) \Rightarrow \varphi(y) \leftrightarrow \varphi(\min u\,.\,\varphi(u))$	**TR** $2, 3, 4$
6.	$\varphi(y) \Rightarrow \varphi(\min u\,.\,\varphi(u))$	**TR** 5

∎

Lemma 6 *Let $\varphi(y)$ be uniquely extensible. Then*

$$\vdash \exists y :\ \Box\, \varphi(y)$$

Proof:

1.	$\varphi(y) \Rightarrow \varphi(\min u\,.\,\varphi(u))$	Claim 5
2.	$\exists y :\ \varphi(y) \Rightarrow \varphi(\min u\,.\,\varphi(u))$	$\exists$-**INTR** 1
3.	$\Box\, \exists y :\ \varphi(y) \rightarrow \Box\, \varphi(\min u\,.\,\varphi(u))$	**TR** 2
4.	$\Box\, \exists y :\ \varphi(y) \rightarrow \exists z :\ \Box\, \varphi(z)$	QT 3
5.	$\exists z :\ \Box\, \varphi(z)$	**MP** 4, Lemma 3

∎

Let $\varphi(y)$ be an extensible formula. Define

$$\mathit{sext}_{\varphi}(y) : \varphi(y)\ \wedge\ (y = \mathit{min}\ u\,.\,(\varphi(u)\ \wedge\ \widehat{\boxminus}(u = y)))$$

Lemma 7 *Let $\varphi(y)$ be an extensible formula. Then it is provable that*

$$\boxminus\, sext_\varphi(y)$$

is a uniquely extensible formula.

Proof: We have to show the following:

(a)	$\vdash \boxminus(y = z) \Rightarrow (\boxminus\, sext_\varphi(y) \leftrightarrow \boxminus\, sext_\varphi(z))$	past-dependence
(b)	$\vdash \widehat{\boxminus} sext_\varphi(y) \Rightarrow \exists z : \widehat{\boxminus}(y = z) \wedge \boxminus\, sext_\varphi(z)$	extensibility
(c)	$\vdash \boxminus\, sext_\varphi(y) \wedge \boxminus\, sext_\varphi(z) \Rightarrow (y = z)$	unique extensibility
(d)	$\vdash \boxminus\, sext_\varphi(y) \Leftrightarrow \boxminus \boxminus\, sext_\varphi(y)$	unique extensibility

Entailment (a) is a direct result of

$$\vdash \boxminus(y = z) \Rightarrow \boxminus(\varphi(y) \leftrightarrow \varphi(z)).$$

which is easily provable for a past-dependent formula φ.

∎

Proof of (b):

1.	$\widehat{\boxminus} sext_\varphi(y) \Rightarrow \widehat{\boxminus}\varphi(y) \wedge \widehat{\boxminus}(y = \min u\,.\,\varphi(u) \wedge \widehat{\boxminus}(u = y))$	Def. of $sext_\varphi$, **TR**
2.	$\widehat{\boxminus} sext_\varphi(y) \Rightarrow \exists z : \varphi(z) \wedge \widehat{\boxminus}(z = y)$	extensibility of φ
3.	$\underbrace{\varphi(z) \wedge \widehat{\boxminus}(z = y)}_{\psi(z)} \Rightarrow$	
	$\exists w : \varphi(w) \wedge \widehat{\boxminus}(w = y) \wedge (w = \min u\,.\,(\varphi(u) \wedge \widehat{\boxminus}(u = y)))$	Lemma 2 on $\psi(z)$, **TR**
4.	$\exists z : \varphi(z) \wedge \widehat{\boxminus}(z = y) \Rightarrow \exists w : sext_\varphi(w) \wedge \widehat{\boxminus}(w = y)$	$\exists$-**INTR** 3, def. of $sext_\varphi$
5.	$\widehat{\boxminus} sext_\varphi(y) \Rightarrow \exists w : sext_\varphi(w) \wedge \widehat{\boxminus}(w = y)$	**TR** 2, 4
6.	$\widehat{\boxminus}(w = y) \wedge \widehat{\boxminus} sext_\varphi(y) \Rightarrow \widehat{\boxminus} sext_\varphi(w)$	part (a), **TR**
7.	$\widehat{\boxminus} sext_\varphi(y) \wedge sext_\varphi(w) \wedge \widehat{\boxminus}(w = y) \Rightarrow \boxminus\, sext_\varphi(w) \wedge \widehat{\boxminus}(w = y)$	**TR** 6
8.	$\widehat{\boxminus} sext_\varphi(y) \wedge \exists w : sext_\varphi(w) \wedge \widehat{\boxminus}(w = y) \Rightarrow$	
	$\exists w : \boxminus\, sext_\varphi(w) \wedge \widehat{\boxminus}(w = y)$	$\exists\exists$-**INTR** 7
9.	$\widehat{\boxminus} sext_\varphi(y) \Rightarrow \exists w : \boxminus\, sext_\varphi(w) \wedge \widehat{\boxminus}(w = y)$	**TR** 5, 8

∎

Proof of (c): Using induction theorem **NFX**5, we prove the stronger entailment:

$$\vdash \boxminus\, sext_\varphi(y) \wedge \boxminus\, sext_\varphi(z) \Rightarrow \boxminus(y = z)$$

We have to show the following:

$$\vdash \underbrace{(\widehat{\boxminus} sext_\varphi(y) \wedge \widehat{\boxminus} sext_\varphi(z) \rightarrow \widehat{\boxminus}(y = z))}_{p} \Rightarrow (\boxminus\, sext_\varphi(y) \wedge \boxminus\, sext_\varphi(z) \rightarrow \boxminus(y = z))$$

1.	$\widehat{\boxminus}(y = z) \Rightarrow (\min u\,.\,\varphi(u) \wedge \widehat{\boxminus}(u = y) = \min u\,.\,\varphi(u) \wedge \widehat{\boxminus}(u = z))$	**TR**
2.	$sext_\varphi(y) \wedge sext_\varphi(z) \wedge \widehat{\boxminus}(y = z) \Rightarrow (y = z)$	1, Def. of $sext_\varphi$
3.	$p \Rightarrow (\boxminus\, sext_\varphi(y) \wedge \boxminus\, sext_\varphi(z) \rightarrow \boxminus(y = z))$	**TR** 2, Def of p

Provability of congruence (d) is a direct result of the idempotence of the $\boxminus$ operator and PTL completeness.

Lemma 8 *Let $\varphi(y)$ be an extensible formula. Then*

$$\vdash \exists y : \Box\, \varphi(y)$$

Proof:

1.	$\boxminus\, sext_{\varphi}(y) \Rightarrow \varphi(y)$	Def. of $sext_{\varphi}(y)$
2.	$\Box \boxminus\, sext_{\varphi}(y) \rightarrow \Box\, \varphi(y)$	TR 1
3.	$\exists y : \Box \boxminus\, sext_{\varphi}(y) \rightarrow \exists y : \Box\, \varphi(y)$	$\exists\exists$-INTR 2
4.	$\exists y : \Box \boxminus\, sext_{\varphi}(y)$	Lemmas 6, 7
5.	$\exists y : \Box\, \varphi(y)$	MP 3, 4

Let $step(t)$ represent the following QPTL formula

$$step(t) \;:\; \boxminus\, t \wedge \widehat{\Box} \neg t$$

Claim 9 $\quad \vdash \Box\, \exists t : step(t)$

Proof: Using the induction theorem NFX5, we have to show

(a) $\vdash \exists t : step(t)$
(b) $\vdash \ominus\, \exists t : step(t) \Rightarrow \exists t : step(t)$

Proof of (a):

1.	$\boxminus\, first \wedge \widehat{\Box} \neg first$	TR
2.	$step(first)$	Def. of $step$, *1*
3.	$\exists t : step(t)$	QT 2

Proof of (b):

1.	$\ominus(\boxminus\, y \wedge \widehat{\Box} \neg y) \Rightarrow (\boxminus \ominus y \wedge \widehat{\Box} \ominus \neg y)$	TR
2.	$\ominus\, step(y) \Rightarrow step(\ominus\, y)$	Def. of $step$, *1*
3.	$\ominus\, step(y) \Rightarrow \exists t : step(t)$	QT 2
4.	$\exists t : \ominus\, step(t) \Rightarrow \exists t : step(t)$	$\exists$-INTR , 3
5.	$\ominus\, \exists t : step(t) \Rightarrow \exists t : step(t)$	$\exists - \text{COM} - \ominus$ 4

Let g_1, g_2, and g_3 be three expressions, and z be a variable that does not occur free in any of these expressions. Then,

Claim 10 $\vdash \Box\, \exists z : (\widehat{\boxminus}(z = g_1) \wedge (z = g_2) \wedge \widehat{\Box}(z = g_3))$

Proof: Define $\psi(z) : \widehat{\boxminus}(z = g_1) \wedge (z = g_2) \wedge \widehat{\Box}(z = g_3)$.

1.	$step(t)$ $\Rightarrow \psi($ **if** $\bigcirc t$ **then** g_1 **else-if** $t \wedge \bigcirc \neg t$ **then** g_2 **else** $g_3)$	TR, def. of *step and* ψ
2.	$step(t) \Rightarrow \exists z : \psi(z)$	QT 1
3.	$\exists t : step(t) \Rightarrow \exists z : \psi(z)$	$\exists$-INTR 2
4.	$\Box(\exists t : step(t) \rightarrow \exists z : \psi(z))$	TR 3
5.	$\Box\, \exists t : step(t) \rightarrow \Box\, \exists z : \psi(z)$	TR 4
6.	$\Box\, \exists z : (\widehat{\boxminus}(z = g_1) \wedge (z = g_2) \wedge \widehat{\Box}(z = g_3))$	MP 5, Claim 9

∎

Lemma 11 *let $h(y)$ be a historic function. Then*

$\varphi(y) : y = h(y)$

is provably an extensible formula.

Proof: We have to show the following:

(a)	$\vdash \boxminus(y = z) \Rightarrow \varphi(y) \leftrightarrow \varphi(z)$	past-dependence
(b)	$\vdash \circledcirc\, \varphi(y) \Rightarrow \exists z : \varphi(z) \wedge \widehat{\boxminus}(z = y)$	extensibility

Proof of (a):

1.	$\boxminus(y = z) \Rightarrow (y = z) \wedge (h(y) = h(z))$	TR, h historic
2.	$\boxminus(y = z) \Rightarrow (y = h(y)) \leftrightarrow (z = h(z))$	TR 1
3.	$\boxminus(y = z) \Rightarrow \varphi(y) \leftrightarrow \varphi(z)$	2, Def. of φ

▗

Proof of (b): We prove the stronger claim:

$\vdash \Box\, \exists z : \varphi(z) \wedge \widehat{\boxminus}(z = y)$

1.	$\widehat{\boxminus}(z = y) \wedge (z = h(y)) \Rightarrow \varphi(z) \wedge \widehat{\boxminus}(z = y)$	h historic
2.	$\exists z : \widehat{\boxminus}(z = y) \wedge (z = h(y)) \Rightarrow \exists z : \varphi(z) \wedge \widehat{\boxminus}(z = y)$	$\exists\exists$-INTR 1
3.	$\Box(\exists z : \widehat{\boxminus}(z = y) \wedge (z = h(y)) \rightarrow \exists z : \varphi(z) \wedge \widehat{\boxminus}(z = y))$	TR 2
4.	$\Box\, \exists z : \widehat{\boxminus}(z = y) \wedge (z = h(y)) \rightarrow \Box\, \exists z : \varphi(z) \wedge \widehat{\boxminus}(z = y)$	TR 3
5.	$\Box\, \exists z : (\widehat{\boxminus}(z = y) \wedge (z = h(y))$	Claim 10
6.	$\Box\, \exists z : \varphi(z) \wedge \widehat{\boxminus}(z = y)$	TR 4, 5

∎

Theorem 1 *Let $h(y)$ be a historic function and $\varphi(y) : y = h(y)$. Then*

$$\vdash \exists y : \Box(y = h(y))$$

Proof: A direct result of Lemma 8 and Lemma 11.

∎

Let $\varphi(y) : y = f(y)$, where $f(y)$ is a prophetic function. We define the following QPTL formula:

$$\psi(y) : \exists t : \mathit{step}(t) \wedge \Box \Diamond \exists u : (\boxminus \varphi(u) \wedge \boxminus(t \rightarrow u = y)) \tag{19}$$

Claim 12 *Let $f(y)$ be a prophetic function. Then*

$$\vdash \boxminus(y = z) \Rightarrow \widehat{\boxminus}(f(y) = f(z))$$

Proof:

1. $\bigcirc(y = z) \Rightarrow f(y) = f(z)$ — f prophetic
2. $\widehat{\boxminus} \bigcirc(y = z) \Rightarrow \widehat{\boxminus}(f(y) = f(z))$ — TR 1
3. $\boxminus(y = z) \Rightarrow \widehat{\boxminus}(f(y) = f(z))$ — TR 2

∎

Claim 13 *Let $f(u)$ be a prophetic function. Then*

$$\vdash \Box \forall v \exists u : \bigcirc(u = v) \wedge \boxminus(u = f(u))$$

Proof: Using the induction theorem NFX5, we need to show the following:

(a) $\vdash \forall v \exists u : \bigcirc(u = v) \wedge u = f(u)$
(b) $\vdash \ominus \forall v \exists u : \bigcirc(u = v) \wedge \boxminus(u = f(u)) \Rightarrow \forall z \exists w : \bigcirc(w = z) \wedge \boxminus(w = f(w))$

Proof of a:

1.	$\exists u : \bigcirc(u = v) \wedge u = f(v)$	Claim 10, TR
2.	$\bigcirc(u = v) \wedge (u = f(v)) \Rightarrow \bigcirc(u = v) \wedge (u = f(u))$	prophetic func. def.
3.	$\exists u : \bigcirc(u = v) \wedge (u = f(v)) \Rightarrow \exists u : \bigcirc(u = v) \wedge (u = f(u))$	$\exists\exists$-INTR 2
4.	$\exists u : \bigcirc(u = v) \wedge u = f(u)$	TR 1, 3
5.	$\forall v \exists u : \bigcirc(u = v) \wedge u = f(u)$	$\forall$-INTR 4

⌟

Proof of b:

1.	$\ominus \forall v \exists u : \bigcirc(u = v) \wedge \boxminus(u = f(u))$	
	$\Rightarrow \ominus \exists u : \bigcirc(u = f(z)) \wedge \boxminus(u = f(u))$	**Q**1, **Q**2
	$\Rightarrow \exists u : (u = f(z)) \wedge \widehat{\boxminus}(u = f(u))$	**TR**
2.	$(u = f(z)) \wedge \widehat{\boxminus}(u = f(u)) \wedge \widehat{\boxminus}(w = u) \wedge (w = f(z)) \wedge \widehat{\Box}(w = z)$	
	$\Rightarrow \bigcirc(w = z) \wedge \boxminus(w = f(w))$	**TR**, f prophetic
3.	$(u = f(z)) \wedge \widehat{\boxminus}(u = f(u)) \wedge \exists w : \widehat{\boxminus}(w = u) \wedge (w = f(z)) \wedge \widehat{\Box}(w = z)$	
	$\Rightarrow \exists w : \bigcirc(w = z) \wedge \boxminus(w = f(w))$	$\exists\exists$-**INTR** 2
4.	$\Box \exists w : \widehat{\boxminus}(w = u) \wedge (w = f(z)) \wedge \widehat{\Box}(w = z)$	Claim 10
5.	$(u = f(z)) \wedge \widehat{\boxminus}(u = f(u)) \Rightarrow \exists w : \bigcirc(w = z) \wedge \boxminus(w = f(w))$	**TR** 3, 4
6.	$\exists u : (u = f(z)) \wedge \widehat{\boxminus}(u = f(u)) \Rightarrow \exists w : \bigcirc(w = z) \wedge \boxminus(w = f(w))$	$\exists$-**INTR** 5
7.	$\ominus \forall v \exists u : \bigcirc(u = v) \wedge \boxminus(u = f(u)) \Rightarrow \exists w : \bigcirc(w = z) \wedge \boxminus(w = f(w))$	**TR** 1, 6
8.	$\ominus \forall v \exists u : \bigcirc(u = v) \wedge \boxminus(u = f(u)) \Rightarrow \forall z \exists w : \bigcirc(w = z) \wedge \boxminus(w = f(w))$	$\forall$-**GEN** 7

∎

Corollary 14 $\quad \vdash \Box \exists u : \boxminus(u = f(u))$

Lemma 15 *Let $f(y)$ be a prophetic function and $\varphi(y) : y = f(y)$. Then the formula ψ as defined in* (19) *is provably an extensible formula.*

Proof: We have to prove the following:

a.	$\vdash \boxminus(y = z) \Rightarrow \psi(y) \leftrightarrow \psi(z)$	past-dependence
b.	$\vdash \circledcirc \psi(y) \Rightarrow \exists z : \psi(z) \wedge \widehat{\boxminus}(y = z)$	extensibility

Proof of (a): We prove the following

$$\vdash \boxminus(y = z) \wedge \psi(y) \Rightarrow \psi(z)$$

We use the following notation:

$$\chi(y) : \varphi(u) \wedge t \rightarrow u = y$$

1.	$\boxminus(y = z) \wedge \boxminus t \wedge \widehat{\Box} \neg t \Rightarrow \Box \boxminus(t \rightarrow y = z)$	**TR**
2.	$\boxminus(t \rightarrow y = z) \wedge \boxminus \chi(y) \Rightarrow \boxminus \chi(z)$	**TR**
3.	$\Box \boxminus(t \rightarrow y = z) \wedge \Box \Diamond \exists u : \boxminus \chi(y)$	
	$\Rightarrow \Box \Diamond(\boxminus(t \rightarrow y = z) \wedge \exists u : \boxminus \chi(y))$	**TR**
	$\Rightarrow \Box \Diamond \exists u : (\boxminus(t \rightarrow y = z) \wedge \boxminus \chi(y))$	**QR**
	$\Rightarrow \Box \Diamond \exists u : \boxminus \chi(z)$	**TR** 2
4.	$\boxminus(y = z) \wedge \mathit{step}(t) \wedge \Box \Diamond \exists u : \boxminus \chi(y)$	
	$\Rightarrow \mathit{step}(t) \wedge \Box \Diamond \exists u : \boxminus \chi(z)$	**TR** 1, 3
5.	$\boxminus(y = z) \wedge \exists t : \mathit{step}(t) \wedge \Box \Diamond \exists u : \boxminus \chi(y)$	
	$\Rightarrow \exists t : \mathit{step}(t) \wedge \Box \Diamond \exists u : \boxminus \chi(z)$	$\exists\exists$-**INTR** 4
6.	$\boxminus(y = z) \wedge \psi(y) \Rightarrow \psi(z)$	5, Def. of ψ

Proof of (b): We have to prove the following

$$\vdash \circledcirc\, \psi(y) \Rightarrow \exists z : \psi(z) \wedge \widehat{\boxminus}(y = z)$$

We break the proof into two parts, as follows:

Part-1 : $\vdash \mathit{first} \rightarrow \exists z : \psi(z)$
Part-2 : $\vdash \ominus\, \psi(y) \Rightarrow \exists z : \psi(z) \wedge \widehat{\boxminus}(y = z)$

Proof of Part-1: We have to show

$$\vdash \exists t, z : \mathit{step}(t) \wedge \Box \Diamond \exists u : (\boxminus\, \varphi(u) \wedge \boxminus(t \rightarrow u = z))$$

Take $t = \mathit{first}$, and show

$a1$. $\vdash \mathit{step}(\mathit{first})$
$b1$. $\vdash \exists z : \Box \Diamond \exists u : (\boxminus\, \varphi(u) \wedge \boxminus(\mathit{first} \rightarrow u = z))$

$\vdash \mathit{step}(\mathit{first})$ requires $\vdash (\boxminus\, \mathit{first} \wedge \bigcirc \Box \neg \mathit{first})$ which follows from PTL completeness.

Proof of (b1):

1.	$\Box \diamondminus(\mathit{first} \wedge \bigvee_{i=1}^{n} u = k_i)$	**TR**, TAU
2.	$\boxminus\, \varphi(u) \Rightarrow \boxminus\, \varphi(u) \wedge \diamondminus(\mathit{first} \wedge \bigvee_{i=1}^{n} u = k_i)$	**TR** 1
3.	$\Box \exists u : \boxminus\, \varphi(u) \rightarrow \Box \exists u : (\boxminus\, \varphi(u) \wedge \diamondminus(\mathit{first} \wedge \bigvee_{i=1}^{n} u = k_i))$	$\exists\exists$-**INTR** 2, **TR**
4.	$\Box \exists u : \boxminus\, \varphi(u)$	Corollary 14
5.	$\Box \exists u : (\boxminus\, \varphi(u) \wedge \diamondminus(\mathit{first} \wedge \bigvee_{i=1}^{n} u = k_i))$	**MP** 3, 4
6.	$\Box \Diamond \exists u : (\boxminus\, \varphi(u) \wedge \diamondminus(\mathit{first} \wedge \bigvee_{i=1}^{n} u = k_i))$	**TR** 5
7.	$\bigvee_{i=1}^{n} \Box \Diamond \exists u : (\boxminus\, \varphi(u) \wedge \diamondminus(\mathit{first} \wedge u = k_i))$	Distributivity of $\vee$[4]
8.	$\bigvee_{i=1}^{n} \Box \Diamond \exists u : (\boxminus\, \varphi(u) \wedge \boxminus(\mathit{first} \rightarrow u = k_i))$	**TR** 7
9.	$\bigvee_{i=1}^{n} \exists z : \Box \Diamond \exists u : (\boxminus\, \varphi(u) \wedge \boxminus(\mathit{first} \rightarrow u = z))$	QT 8
10.	$\exists z : \Box \Diamond \exists u : (\boxminus\, \varphi(u) \wedge \boxminus(\mathit{first} \rightarrow u = z))$	**TR** 9

Proof of Part-2: We use the following notation

$\eta \quad : \quad \circledcirc\, t \wedge \neg t$

$\vartheta(y,t) \quad : \quad \boxminus\, \varphi(u) \wedge \boxminus(t \rightarrow u = y)$

1. $\diamondminus\, \eta \Rightarrow \diamondminus(\eta \wedge \bigvee_i (u = k_i))$ — **TAU**, **TR**

2. $\exists t : \ominus\, step(t) \wedge \Box \Diamond \exists u : \vartheta(y,t) \wedge \diamondminus\, \eta$
$\Rightarrow \exists t : \ominus\, step(t) \wedge \Box \Diamond \exists u : (\vartheta(y,t) \wedge \diamondminus(\eta \wedge \bigvee_i u = k_i))$ — **TR**, 1, $\exists\exists$-**INTR**

3. $\diamondminus\, \eta \wedge \exists u : \vartheta(y,t) \Rightarrow \exists u : (\vartheta(y,t) \wedge \diamondminus\, \eta)$ — $u \notin \eta$

4. $\ominus\, step(t) \Rightarrow \Box \diamondminus\, \eta$ — **TR**, defs of $step, \eta$

5. $\Box \diamondminus\, \eta \wedge \Box \Diamond \exists u : \vartheta(y,t) \Rightarrow \Box \Diamond(\diamondminus\, \eta \wedge \exists u : \vartheta(y,t))$ — **TR**

6. $\Box \Diamond(\diamondminus\, \eta \wedge \exists u : \vartheta(y,t)) \Rightarrow \Box \Diamond \exists u : (\vartheta(y,t) \wedge \diamondminus\, \eta)$ — **TR** 3

7. $\ominus\, step(t) \wedge \Box \Diamond \exists u : \vartheta(y,t)$
$\Rightarrow \ominus\, step(t) \wedge \Box \Diamond \exists u : (\vartheta(y,t) \wedge \diamondminus\, \eta)$ — **TR** 4, 5, 6

8. $\exists t : \ominus\, step(t) \wedge \Box \Diamond \exists u : \vartheta(y,t)$
$\Rightarrow \exists t : \ominus\, step(t) \wedge \Box \Diamond \exists u : (\vartheta(y,t) \wedge \diamondminus\, \eta)$ — $\exists\exists$-**INTR** 7

9. $\ominus\, \psi(y) \Rightarrow \exists t : \ominus\, step(t) \wedge \Box \Diamond \exists u : (\vartheta(y,t) \wedge \diamondminus(\eta \wedge \bigvee_i u = k_i))$ — **TR** 8, 2

10. $\ominus\, \psi(y) \Rightarrow \bigvee_i \exists t : \underbrace{\ominus\, step(t) \wedge \Box \Diamond \exists u : \vartheta(y,t) \wedge \diamondminus(\eta \wedge (u = k_i))}_{q_i}$ — Distributivity of $\vee$ [4]

11. $\ominus\, step(t) \wedge \underbrace{z = k_i \wedge \widehat{\boxminus}(z = y)}_{p}$
$\Rightarrow \Box(\vartheta(y,t) \wedge \diamondminus(\eta \wedge u = k_i) \rightarrow \vartheta(z, \ominus\, t))$ — **TR**

12. $\ominus\, step(t) \wedge p \Rightarrow \Box \forall u : (\vartheta(y,t) \wedge \diamondminus(\eta \wedge u = k_i) \rightarrow \vartheta(z, \ominus\, t))$ — $\forall$-**INTR** 11, **TR**

13. $p \wedge q_i \Rightarrow \left[\begin{array}{c} \Box \Diamond \exists u : \vartheta(y,t) \wedge \diamondminus(\eta \wedge u = k_i) \\ \wedge \\ \Box \forall u : (\vartheta(y,t) \wedge \diamondminus(\eta \wedge u = k_i) \rightarrow \vartheta(z, \ominus\, t)) \end{array}\right]$ — **TR** 12

[4] Note that $\vee$ does not distribute over $\Box$ but distributes over the combination $\Box \Diamond$. Note also that $\Box \Diamond$ distributes over $\vee$ but not over $\exists$, which explains why provability is restricted to finite domains.

14.	$p \wedge q_i \Rightarrow \Box \Diamond \exists u : \left[\begin{array}{c} \vartheta(y,t) \wedge \diamondminus(\eta \wedge u = k_i) \\ \wedge \\ \vartheta(y,t) \wedge \diamondminus(\eta \wedge u = k_i) \rightarrow \vartheta(z, \ominus t)) \end{array}\right]$	TR, QR 13
15.	$p \wedge q_i \Rightarrow \Box \Diamond \exists u : \vartheta(z, \ominus t)$	TR 14
16.	$p \wedge q_i \Rightarrow step(\ominus t) \wedge \Box \Diamond \exists u : \vartheta(z, \ominus t) \wedge \widehat{\boxminus}(z = y)$	15, def. of p, q_i
17.	$p \wedge q_i \Rightarrow \exists t : step(t) \wedge \Box \Diamond \exists u : \vartheta(z,t) \wedge \widehat{\boxminus}(z = y)$	QT 16
18.	$\exists t : q_i \wedge \exists z : (z = k_i \wedge \widehat{\boxminus}(z = y)) \Rightarrow \exists z : \psi(z) \wedge \widehat{\boxminus}(z = y)$	$\exists$-INTR , $\exists\exists$-INTR 17, p def.
19.	$\Box \exists z : z = k_i \wedge \widehat{\boxminus}(z = y)$	Claim 10
20.	$\exists t : q_i \Rightarrow \exists z : \psi(z) \wedge \widehat{\boxminus}(z = y)$	TR 18, 19
21.	$\bigvee_i \exists t : q_i \Rightarrow \exists z : \psi(z) \wedge \widehat{\boxminus}(z = y)$	TR 20
22.	$\ominus \psi(y) \Rightarrow \exists z : \psi(z) \wedge \widehat{\boxminus}(z = y)$	TR 10, 21

∎

Lemma 16 *Let $f(y)$ be a prophetic function and $\varphi(y) : y = f(y)$. Then*

$\vdash \psi(y) \Rightarrow \widehat{\boxminus}(\varphi(y))$

Proof:

$\exists t : \boxminus t \wedge \widehat{\Box} \neg t \wedge \Box \Diamond \exists u : (\boxminus \varphi(u) \wedge \boxminus(t \rightarrow u = y))$	
$\Rightarrow \exists t : \boxminus t \wedge \widehat{\Box} \neg t \wedge \Diamond \exists u : (\boxminus \varphi(u) \wedge \boxminus(t \rightarrow u = y))$	TR
$\Rightarrow \exists t, u : \boxminus t \wedge \widehat{\Box} \neg t \wedge \Diamond(\boxminus \varphi(u) \wedge \boxminus(t \rightarrow u = y))$	Commutativuty of $\exists, \Diamond$
$\Rightarrow \exists t, u : \boxminus t \wedge \widehat{\Box} \neg t \wedge \boxminus \varphi(u) \wedge \boxminus(t \rightarrow u = y)$	TR
$\Rightarrow \exists u : \boxminus \varphi(u) \wedge \boxminus(y = u)$	TR
$\Rightarrow \widehat{\boxminus} \varphi(y)$	$f$prophetic + Claim 12

∎

Theorem 3 *Let $f(y)$ be a prophetic function and $\varphi(y) : y = f(y)$. Then*

$\vdash \exists y : \Box \varphi(y)$

Proof:

1.	$\psi(y) \Rightarrow \widehat{\boxminus} \varphi(y)$	Lemma 16
2.	$\Box \psi(y) \rightarrow \Box \widehat{\boxminus} \varphi(y)$	TR 1
3.	$\Box \psi(y) \rightarrow \Box \varphi(y)$	TR 2
4.	$\exists y : \Box \psi(y) \rightarrow \exists y : \Box \varphi(y)$	$\exists\exists$-INTR 3
5.	$\exists y : \Box \psi(y)$	Lemma 8, Lemma 15
6.	$\exists y : \Box \varphi(y)$	MP 4, 5

References

[Aba89] M. Abadi. The power of temporal proofs. *Theor. Comp. Sci.*, 65:35–84, 1989.

[AL91] M. Abadi and L. Lamport. The existence of refinement mappings. *Theoretical Computer Science*, 82(2):253–284, May 1991.

[ANS79] H. Andreka, I. Nemeti, and I. Sain. Completeness problems in verification of programs and program schemes. In *4th Mathematical Foundations of Computer Science (MFCS 79)*, volume 74 of *Lect. Notes in Comp. Sci.*, pages 208–218. Springer-Verlag, 1979. Invited Lecture.

[BB86] B. Banieqbal and H. Barringer. A study of an extended temporal logic and a temporal fixed point calculus. Technical report, University of Manchester, UMCS-86-10-2, 1986.

[Buc62] J.R. Buchi. On a decision method in resricted second-order arithmetics. In *Proc. Int'r Congr. Logic, Method. Phil. of Sci., 1960*, pages 1–12. Stanford University Press, 1962.

[Coo78] S.A. Cook. Soundness and completeness of an axiom system for program verification. *SIAM J. Comp.*, 7:70–90, 1978.

[CY95] C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *J. ACM*, 42:857–907, 1995.

[Gab87] D. Gabbay. The declarative past and imperative future. In B. Banieqbal, H. Barringer, and A. Pnueli, editors, *Temporal Logic in Specification*, volume 398 of *Lect. Notes in Comp. Sci.*, pages 407–448. Springer-Verlag, 1987.

[GPSS80] D. Gabbay, A. Pnueli, S. Shelah, and J. Stavi. On the temporal analysis of fairness. In *Proc. 7th ACM Symp. Princ. of Prog. Lang.*, pages 163–173, 1980.

[Hod95] I. Hodkinson. On Gabbay's Temporal Fixed Point Operator. *Theor. Comp. Sci.*, 139:1–25, 1995.

[HWZ00] I. Hodkinson, F. Wolter, and M. Zakharyaschev. Decidable fragments of first-order temporal logics. *Annals of Pure and Applied Logic*, 106:85–134, 2000.

[Jon87] B. Jonsson. *Compositional Verification of Distributed Systems.* PhD thesis, Uppsala University, Sweden, 1987.

[Kam68] J.A.W. Kamp. *Tense Logic and the Theory of Order.* PhD thesis, UCLA, 1968.

[Kla91] N. Klarlund. Progress measures for complementation of ω-automata with applications to temporal logic. In *Proc. 32nd FOCS*, pages 358–367, 1991.

[KMP94] Y. Kesten, Z. Manna, and A. Pnueli. Temporal verification of simulation and refinement. In J.W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors, *A Decade of Concurrency*, volume 803 of *Lect. Notes in Comp. Sci.*, pages 273–346. Springer-Verlag, 1994.

[KV97] O. Kupferman and M. Y. Vardi. Weak alternating automata are not that weak. In *Proc. 5th Israeli Symp. on Theory of Computing and Systems*, pages 147–158, 1997.

[Lam83] L. Lamport. Specifying concurrent program modules. *ACM Trans. Prog. Lang. Sys.*, 5:190–222, 1983.

[Lam94] L. Lamport. The temporal logic of actions. *ACM Trans. Prog. Lang. Sys.*, 16(3):872–923, May 1994.

[Lic91] O. Lichtenstein. *Decidability, Completeness, and Extensions of Linear Time Temporal Logic.* PhD thesis, The Weizmann Institute if Science, Israel, 1991.

[LP00] O. Lichtenstein and A. Pnueli. Propositional temporal logics: Decidability and completeness. *Logic Journal of the IGPL*, 8(1):1–31, 2000.

[LPZ85] O. Lichtenstein, A. Pnueli, and L. Zuck. The glory of the past. In *Proc. Conf. Logics of Programs*, volume 193 of *Lect. Notes in Comp. Sci.*, pages 196–218. Springer-Verlag, 1985.

[LS84] S.S. Lam and A.U. Shankar. Protocol verification via projections. *IEEE Trans. Software Engin.*, 10(4):325–342, 1984.

[LT87] N. Lynch and M. Tuttle. Hierarchical correctness proofs for distributed algorithms. In *Proc. 6th ACM Symp. Princ. of Dist. Comp.*, pages 137–151, 1987.

[McN66] R. McNaughton. Testing and generating infinite sequences by a finite automaton. *Inf. and Cont.*, 9:521–530, 1966.

[Mic88] M. Michel. Complementation is more difficult with automata on infinite words. Manuscript, 1988.

[MP71] R. McNaughton and S. Papert. *Counter Free Automata.* MIT Press, 1971.

[MP83] Z. Manna and A. Pnueli. How to cook a temporal proof system for your pet language. In *Proc. 10th ACM Symp. Princ. of Prog. Lang.*, pages 141–154, 1983.

[MP91] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification.* Springer Verlag, New York, 1991.

[MS73] A.R. Meyer and L.J. Stockmeyer. Non-elementary word problems in automata and logic. In *Proc. AMS Symp. on Complexity of Computation*, April 1973.

[Saf88] S. Safra. On the complexity of ω-automata. In *Proc. 29th IEEE Symp. Found. of Comp. Sci.*, pages 319–327, 1988. An extended version to appear in *J. Comp. Sys. Sci.*

[Sie70] D. Siefkes. *Decidable Theories I — Büchi's Monadic second-order successor arithmetics.* Lec. Notes Math. 120, Springer-Verlag, 1970.

[SVW87] A.P. Sistla, M.Y. Vardi, and P. Wolper. The complementation problem for Büchi autamata with application to temporal logic. *Theor. Comp. Sci.*, 49:217–237, 1987.

[Sza86] A. Szalas. Concerning the semantic consequence relation in first-order temporal logic. *Theor. Comp. Sci.*, 47:329–334, 1986.

[Wol83] P. Wolper. Temporal logic can be more expressive. *Inf. and Cont.*, 56:72–99, 1983.

[WZ00] F. Wolter and M. Zakharyaschev. Axiomatizing the monodic fragment of first-order temporal logic. 2000. Submitted.