Abstract

We present a natural deduction calculus for the quantified propositional linear-time temporal logic (QPTL) and prove its correctness. The system extends previous natural deduction constructions for the propositional linear-time temporal logic. These developments extend the applicability of the natural deduction to more sophisticated specifications due to the expressive power of QPTL and, on the hand, supply QPTL itself with an elegant reasoning tool.

1 Introduction

In this paper we continue our investigation of natural deduction framework for non-classical setting, this time tackling propositional linear-time temporal logic extended with propositional quantification [22].

Informally, propositional quantification means the introduction of the so called ‘second order propositional logic’, where we can express some knowledge about propositions by means of the expressions of the type $\exists x . A$ and $\forall x . A$ – ‘for some proposition $x$, $A$’ and ‘for any proposition $x$, $A$’.

Some analysis of the propositional quantification goes back to Russel [20] and Church [7]. It is known that in the setting of classical logic we have a very intuitive interpretation of the propositional quantification simply due to the fact that every proposition is true ($\top$) or false ($\bot$). Thus, $\exists x . A$ states that there is an interpretation for $x$ which makes $A$ true, i.e. $\exists x . A = A(x/\top) \lor A(x/\bot)$.

Similarly, $\forall x . A$ states that any interpretation of $x$ makes $A$ true, i.e. $\forall x . A = A(x/\top) \land A(x/\bot)$, where $A(x/\top)$ means a formula obtained from $A$ by the substitution of all occurrences of $x$ in $A$ by $\top$. Therefore, propositional quantification does not add any expressiveness to the classical logic.

Things are different when we add propositional quantification to a non-classical propositional logic. Here we emphasize the following two main directions of exploring the addition of the propositional quantifiers. Firstly, it is the QSAT framework, i.e. SAT for the quantified Boolean formulae [18], which is important, for example, from the point of view of its use in model checking. Secondly, it is the enriched expressiveness gained through the extensions of various non-classical propositional logics with the propositional quantification. For example, the framework of propositional quantification in modal logic was initially studied in [6, 9].

The setting of the propositional quantification in the propositional linear time temporal logic (PLTL) was introduced in [22]. We follow the notation adopted in [12] calling this logic QPTL. Thus, QPTL is more expressive than PLTL presenting the same potential of expressiveness as linear-time $\mu$-calculus (linear-time propositional temporal fixpoint logic) [15], ETL (propositional linear-time temporal logic extended with automata constraints) [23] and S1S (second order logic of one successor) [15] so that each of these formalisms is as expressive as Buchi Automata [5]. A well-known example distinguishing the expressiveness of these formalisms comparing to PLTL is their ability to ”count”, for example, to express that some property occurs at every even moment of time [23]. Nevertheless, each of these logics uses its own specific syntax and it makes sense to consider how easy these logics can be used in specification. We believe that in this list QPTL indeed occupies a special place. For example, ETL and linear time $\mu$-calculus formulae are very difficult for understanding. It is known,
in particular, that formulae with nested fixpoints very quickly become incomprehensible while automata constraints added to the logic are far too complex to apprehend intuitively.

To the best of our knowledge the only deductive systems for QPTL were introduced in [16] (for QPTL with both future and past temporal operators) and in [12] (for QPTL defined only in terms of future temporal operators). In this paper we present a sound and complete natural deduction based proof theory for QPTL defined with the future operators only.

As QPTL combines the propositional linear time logic with propositional quantification, we work, on the one hand, in our old framework of natural deduction calculi originally developed by Gentzen [13] and Jaskowski [14] and improved by Fitch [10] and Quine [19]. On the other hand, we build upon our previous ND constructions for the logic PLTL [1] and first order logic [3]. Namely, the rules for the linear-time framework are adopted from the former paper while the ideology for the rules for the propositional quantifiers are taken from the latter. Our construction naturally extends previously defined system for PLTL [1] by new rules to capture the propositional quantification in the setting of linear time. The new rules, for the propositional quantifiers, allow us to decompose formulae eliminating either the \( \forall \) (freely) or \( \exists \) (with some obvious restrictions) propositional quantifier from the formula or, on the contrary, to synthesise formulae introducing these quantifiers: \( \forall \) (with some restrictions) or \( \exists \) (without restrictions). To capture other specific QPTL properties, such as quantified induction, we introduce a range of new rules, such as for example, a 0-premise rule \( \vdash \exists x.\).

Note, however, that though our ND system for the logic QPTL is based on the previous developments it has not been obtained from them straightforwardly and automatically. The introduction of the new rules for the propositional quantification and the correctness arguments for the new system were obtained through the careful investigation of important semantical properties of the underlying system (which themselves are far from being trivial). An additional contribution of the paper is in the simplification and refinement of the technical side of the presentation. Many of these aspects, such as the process of discharging assumptions and side conditions on the ND rules related to marking/binding of variables [1], were addressed following the comments of anonymous referees of our earlier (unpublished) work and through the correspondence with some of our readers and communication with the experts in scientific fora.

Our correctness arguments establishes that the introduced system is sound and complete. In proof of completeness we use the methodology of showing the deductive equivalence between our system and the axiomatics for QPTL from [12], i.e. we show that every theorem of the axiomatics is a theorem in our system.

The paper is organized as follows. In §2 we review the syntax and semantics of QPTL. In §3 we describe the ND for QPTL henceforth referred to as QPTL\(_{ND}\) and give an example of the construction of the proof. Subsequently, in §4, we provide the correctness argument. Finally, in §6, we draw concluding remarks and identify future work.

## 2 Syntax and Semantics of QPTL

We define the language of QPTL using the following symbols:

- a set, \( \text{Prop} \), of atomic propositions \( p, q, r, \ldots, p_1, q_1, r_1, \ldots, p_n, q_n, r_n, \ldots \);
- classical operators: \( \neg, \land, \rightarrow, \lor \);
- temporal operators: \( [\mathbf{a}] – \text{‘always in the future’}; [\mathbf{\diamond}] – \text{‘at sometime in the future’}; [\mathbf{\square}] – \text{‘at the next moment in time’}; \) and, finally, propositional quantifiers \( \forall \) (‘for all’) and \( \exists \) (‘there exists’).

The class of well formed QPTL formulae is inductively defined below (where \( \alpha \in \text{Prop} \)).

**Definition 1 (QPTL well-formed formulae)**

1. All atomic propositions (members of \( \text{Prop} \)) are in \( \text{wff}_{\text{QPTL}} \).
2. If \( A \) and \( B \) are in \( \text{wff}_{\text{QPTL}} \), then so are \( A \land B, \neg A, A \lor B, \text{and } A \rightarrow B \).
3. If \( A \) and \( B \) are in \( \text{wff}_{\text{QPTL}} \), then so are \( [\mathbf{a}]A, [\mathbf{\diamond}]A, [\mathbf{\square}]A \), and \( A U B \).
4. If \( A \) is in \( \text{wff}_{\text{QPTL}} \) and \( \alpha \) is in \( \text{Prop} \) then \( \forall \alpha A \) and \( \exists \alpha A \) are in \( \text{wff}_{\text{QPTL}} \).

Note that based on its characterisation as a minimal fixpoint, the \( U \) operator is expressible in QPTL as follows [15]:

\[
\exists z, z \land [\mathbf{\diamond}]z \land [\mathbf{\square}]z \Rightarrow (b \lor (a \land z))
\]

A model, \( \sigma \), for QPTL formulae is defined as follows.
3.1 Extended QPTL Syntax and Semantics

Let \( g \) which can be interpreted as

Definition 2 (QPTL model)
A model for QPTL formulae, is a discrete, linear sequence of states

\[ \sigma = s_0, s_1, s_2, \ldots \]

which satisfies the following properties:

- \( \sigma \) is isomorphic to the natural numbers, \( \mathbb{N} \);
- a state \( s_i \), \( 0 \leq i \), evaluates propositional variables from Prop at the \( i \)-th moment of time, namely, it evaluates a proposition \( \alpha \) as true if \( \alpha \in s_i \), and false otherwise.
- for any two states \( s_i, s_j, i \neq j, 0 \leq i, j \), there exists a variable \( \alpha \in \text{Prop} \) such that \( \alpha \in s_i \) and \( \alpha \notin s_j \).

Note that the last condition simply states that there are no identical, or repeating, states in a model for QPTL. As shown in [11], QPTL interpreted over the structure which allows repeating states is undecidable. We will return to this issue several times later, addressing the effects of the ban on repetitive words on the validity and provability of QPTL formulae.

By \( \langle \sigma, i \rangle \models p \) we understand that in a model \( \sigma \), variable \( p \) is assigned the value true at the moment \( i \) (or contained in \( s_j \)). Given a model \( \sigma \), for a variable \( \alpha \in \text{Prop} \), the model \( \sigma' = s_0', s_1', s_2', \ldots \) is called an \( \alpha \)-variant of \( \sigma \) provided that \( \sigma' \) differs from \( \sigma \) at most in interpreting \( \alpha \). Figure 1 generalises the relation ‘\( \models \)’ to QPTL formulae, where indices \( i, j, k, l \in \mathbb{N} \).

\[
\begin{align*}
\langle \sigma, i \rangle \models p & \quad \text{iff} \quad p \in s_i, \text{ for } p \in \text{Prop} \\
\langle \sigma, i \rangle \models \neg A & \quad \text{iff} \quad \langle \sigma, i \rangle \not\models A \\
\langle \sigma, i \rangle \models A \land B & \quad \text{iff} \quad \langle \sigma, i \rangle \models A \land \langle \sigma, i \rangle \models B \\
\langle \sigma, i \rangle \models A \lor B & \quad \text{iff} \quad \langle \sigma, i \rangle \models A \lor \langle \sigma, i \rangle \models B \\
\langle \sigma, i \rangle \models A \rightarrow B & \quad \text{iff} \quad \langle \sigma, i \rangle \not\models A \lor \langle \sigma, i \rangle \models B \\
\langle \sigma, i \rangle \models \Box A & \quad \text{iff} \quad \text{for each } j \text{ if } i \leq j \text{ then } \langle \sigma, j \rangle \models A \\
\langle \sigma, i \rangle \models \Diamond A & \quad \text{iff} \quad \text{there exists } j \text{ such that } i \leq j \text{ and } \langle \sigma, j \rangle \models A \\
\langle \sigma, i \rangle \models \square A & \quad \text{iff} \quad \langle \sigma, i + 1 \rangle \models A \\
\langle \sigma, i \rangle \models \forall \alpha A & \quad \text{iff} \quad \text{for any } \alpha \text{-variant, } \langle \sigma', i \rangle \models A \\
\langle \sigma, i \rangle \models \exists \alpha A & \quad \text{iff} \quad \text{there exists } \alpha \text{-variant, such that } \langle \sigma', i \rangle \models A
\end{align*}
\]

Figure 1. QPTL Semantics

Definition 3 (QPTL Satisfiability) A well-formed formula, \( A \), is satisfiable if, and only if, there exists a model \( \sigma \) such that \( \langle \sigma, 0 \rangle \models A \).

Definition 4 (QPTL Validity) A well-formed formula, \( A \), is valid if, and only if, \( A \) is satisfied in every possible model, i.e. for each \( \sigma \), \( \langle \sigma, 0 \rangle \models A \).

An interesting example of a valid formula with temporal operators would be

\[
\exists x. x \land \Box \neg x
\]

which can be interpreted as ‘now will never happen again’ [11]. Obviously, this formula is valid only in the semantics without repeating states.

3 Natural Deduction System \( QPTL_{ND} \)

3.1 Extended QPTL Syntax and Semantics

To define the rules of the natural deduction system we extend the syntax of QPTL by introducing labelled formulae.

Firstly, we define the set of labels, \( \text{Lab} \), as a set of variables interpreted over states of \( \sigma \):

\[
\text{Lab} : \{ i, j, k, i_1, j_1, k_1, \ldots \}.
\]

Let \( g \) be a function, which maps the set \( \text{Lab} \) to \( \mathbb{N} \).

We then define two binary relations ‘\( \leq \)’ and ‘Next’, and the operation ‘\( \)’ as follows.
Definition 5 (Relations \(\prec, \simeq, \preceq\) and \(Next\), operation ‘\(\)’) For \(x, y \in Lab\):

(5.1) \(\prec \subseteq \text{Lab}^2 : i \prec j \iff g(i) < g(j)\).
(5.2) \(\simeq \subseteq \text{Lab}^2 : i \simeq j \iff g(i) = g(j)\).
(5.3) \(\preceq \subseteq \text{Lab}^2 : i \preceq j \iff g(i) \leq g(j)\).
(5.4) \(\text{Next} \subseteq \text{Lab}^2 : \text{Next}(i, j) \iff g(j) = g(i) + 1\), i.e. it is the ‘predecessor-successor’ relation such that for any \(i \in \text{Lab}\), there exists \(j \in \text{Lab}\) such that \(\text{Next}(i, j)\) (seriality).
(5.5) Given a label \(i\), the operation ‘\(\)’ applied to \(i\) gives us the label \(i’\) such that \(\text{Next}(i, i’)\).

The following properties follow straightforwardly from Definition 5.

Lemma 1 [Properties of \(\preceq\) and \(\text{Next}\)]

- For any \(i, j \in \text{Lab}\) if \(\text{Next}(i, j)\) then \(i \preceq j\).
- For any \(i, j \in \text{Lab}\) if \(i \prec j\) then \(i \preceq j\).
- Properties of \(\preceq\):
  - For any \(i \in \text{Lab}\) : \(i \preceq i\) (reflexivity),
  - For any \(i, j, k \in \text{Lab}\) if \(i \preceq j\) and \(j \preceq k\) then \(i \preceq k\) (transitivity).

Again, following [21], the expressions representing the properties of \(\preceq\) and \(\text{Next}\) are called ‘relational judgements’. Now we are ready to introduce the \(\text{QPTL}_{ND}\) syntax.

Definition 6 (\(\text{QPTL}_{ND}\) Syntax)

- If \(A\) is a \(\text{QPTL}\) formula and \(i \in \text{Lab}\) then \(\langle i, A \rangle\) is a \(\text{QPTL}_{ND}\) formula.
- Any relational judgement of the type \(\text{Next}(i, i')\) and \(i \preceq j\) is a \(\text{QPTL}_{ND}\) formula.

For the interpretation of \(\text{QPTL}_{ND}\) formulae we adopt the semantical constructions defined in §2 for the logic QPTL. In the rest of the paper we will use capital letters \(A, B, C, D, \ldots\) as metasymbols for QPTL formulae, and calligraphic letters \(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \ldots\) to abbreviate formulae of \(\text{QPTL}_{ND}\), i.e. either labelled formulae or relational judgements. The intuitive meaning of \(\langle i, A \rangle\) is that \(A\) is satisfied at the world \(i\). Thus, based on our observations above, we simply need the following statements.

Let \(\Gamma\) be a set of \(\text{QPTL}_{ND}\) formulae, let \(D_\Gamma = \{\langle i, A \rangle : A \in \Gamma\}\), let \(\sigma\) be a model as defined in §2 and let \(f\) be a function which maps elements of \(D_\Gamma\) into \(\mathcal{N}\) (recall that a QPTL model \(\sigma\) is isomorphic to natural numbers).

Definition 7 (Realisation of \(\text{QPTL}_{ND}\) Formulae in a Model) Model \(\sigma\) realises a set, \(\Gamma\), if there is a mapping, \(f\), which satisfies the following conditions.

1. For any \(i \in D_\Gamma\), and for any \(A\), if \(\langle i, A \rangle \in \Gamma\) then \(\langle \sigma, f(i) \rangle \models A\).
2. For any \(i, j, \) if \(i \preceq j \in \Gamma\) then \(f(i) \leq f(j)\).
3. For any \(i, j, \) if \(\text{Next}(i, j) \in \Gamma\), then \(f(j) = f(i) + 1\).

The set \(\Gamma\) in this case is called realisable.

Definition 8 (\(\text{QPTL}_{ND}\) Validity)

A well-formed \(\text{QPTL}_{ND}\) formula, \(\mathcal{A} = i : B\), is valid (abbreviated as \(\models_{ND} \mathcal{A}\)) if, and only if, the set \(\{\mathcal{A}\}\) is realisable in every possible model, for any function \(f\).

It is easy to see that if we ignore the labels then the classes of satisfiable and valid formulae introduced by definitions 3 and 7, 4 and 8 respectively, are identical.
3.2 QPTL\textsubscript{ND} Rules

Before introducing the rules for the ND system we give some informal description of the ND philosophy and several relevant definitions.

The set of rules is divided into the two classes: elimination and introduction rules. Rules of the first group allow us to simplify formulae to which they are applied. These are rules for the ‘elimination’ of logical constants. Rules of the second group are aimed at ‘building’ formulae, introducing new logical constants. Every ND proof commences with an assumption, and, in general, any formula can be taken as the assumption at any step of the ND proof. However, to prove a theorem we need to have a proof with no assumptions, or, in our setting, we need to discharge assumptions. In our system this discharging is required when we introduce \( \Rightarrow \) or \( \neg \). Of course, it is only the search strategy which guide us through the process of making assumptions, ensuring, in case, of a decidable logic, that all assumptions can be discharged to obtain proofs of theorems.

As mentioned in the introduction we refine here many of the notions contained in our earlier works [1, 4]. Thus, firstly, we provide an explicit procedure to work with the assumptions in our proofs.

Definition 9 (Procedure ‘Assumptions’) We define the construction of the sequence of assumptions, Assumptions, which have been introduced into the proof, \( \pi \), at some step \( n \), as follows:

- The first formula of \( \pi \) (which is always an assumption) is written into Assumptions as its initial component.
- If \( A_i \) \( (0 < i \leq n) \) is a new assumption then update Assumptions by adding \( A_i \) at the end of the sequence Assumptions.
- If \( A_i \) \( (0 < i \leq n) \) has been discharged then update Assumptions by deleting \( A_i \) from it.

The last formula in Assumptions is abbreviated as \( MRA \) (the most recent assumption of \( \pi \)). An index \( \pi \) will be omitted if it is obvious which proof is considered.

Note that when all elements of Assumptions are discharged Assumptions is empty.

Though we preserve the rules for Boolean and temporal operations from our previous work [1], for the consistency of presentation we also repeat these rules here. Thus, in Figure 2 we define the sets of elimination and introduction rules, where prefixes ‘\( \ell \)’ and ‘\( \emph{in} \)’ abbreviate an elimination and an introduction rule, respectively.

In the formulation of the rules ‘\( \Rightarrow \emph{in} \)’ and ‘\( \neg \emph{in} \)’ expressions \( [i:C] \) and \( [j:C] \) abbreviate respectively that formulae \( i:C \) and \( j:C \) must be the MRA. When we apply one of these rules at step \( n \) and discharge an assumption at step \( m \), we also discard all formulae from \( m \) to \( n-1 \). We write \( [m-(n-1)] \) to indicate this situation.

Though it has been shown that these rules are sufficient to prove any classical theorem, we add another rule which is deeply involved into our searching procedure [4]. It simply represents one of De Morgan laws and is derivable from the set of classical rules mentioned above.

\[
\neg \lor \quad i: \neg (A \lor B) \\
\frac{}{i: \neg A \land \neg B}
\]

Figure 3 introduces the rules to manipulate with relational judgements which correspond to the properties introduced by Definition 5.

The linearity rule needs some additional comments. Strictly speaking, in the QPTL\textsubscript{ND} language, to avoid unnecessary complications, we do not allow either Boolean combination of relational judgements or their negations. Obviously, the conclusion of the \( \preceq \) linearity rule violates this constraint. However, it expresses an obvious property of the linear time model structure and to make our presentation more transparent we explicitly formulate a corresponding rule. Our justification here is very simple: the only way in which the conclusion of this rule is involved into the construction of the proof is reasoning by cases, see [4] for details.

Side conditions on temporal rules. Recall that in our previous work [1] we had the notion of binding variables which we called ‘flagging’ following the tradition: when a label, \( j \), is flagged, abbreviated as \( \rightarrow j \), it is bound to a state and, hence, cannot be rebound to some other state. Any label \( i \) paired with the flagged label \( j \) in some relational judgement, say, \( i \preceq j \) becomes relatively flagged (relatively bound) by \( j \), with the meaning that \( j \) restricts the set of runs for \( i \) that is linked to it in the relational judgment. The latter can happen when a we try to derive \( \Box_p \) from two formulae \( i: p \), \( i \preceq i \). Applying
the $\square \in$ rule would require here $i$ to relatively bind itself. This self-binding was, of course, subject of a ban for completed proofs. Note that we reasoned analogously to the predicate logic, where relative binding was a transitive relation, hence, it was possible to have situations when a variable would bind itself by transitivity. However, investigating the ways how this relative binding may affect our proof in temporal setting we realised that we will never have a self binding due to transitivity. Therefore, we have decided to simplify the side conditions on the rule, simply requiring now that in the relational judgements which appear in the rules $\diamond \text{el}$ and $\square \in$, we have a strict $\prec$ relation thus eliminating a possibility of self-binding.

Rules for Propositional Quantification Explained

First of all we utilise the standard notion of substitution: we would say that $A(\beta/\alpha)$ is the result of a correct substitution of a variable $\alpha$ instead of free occurrences of a variable $\beta$ in formula $A$ with the standard condition on the correct substitution.

When applying the rules for the quantifiers we will have to restrict scopes of the variables, similar to the rules for the first-order quantifiers in [3]. When we eliminate the existential quantifier, say from $i : \exists x A(x)$, where $A(x)$ is a propositional formula with some occurrences of $x$, we conclude $i : A(x/y)$, with an obvious restriction on $y$ to be bound in the context of the world $i$.

The rules $\exists_1$ and $\exists_2$ are two 0-premise rules expressing the basic property (basic validities) of the propositional quantification. They are needed for the completeness of the system, in particular, for the proof of the axiom QX0.

Definition 10 (QPTL$_{ND}$ rules for temporal logic operators, propositional quantifiers, and the induction rule)

Figures 4 and 5 provide elimination and introduction rules for the temporal logic operators, new rules for the propositional quantifiers, and the induction rule with the following side conditions:

- Corresponding world variables are bound in the rules $\diamond \text{el}$, $\Box \text{el}$, $\square \in$, and the induction rule.
- In rules $\exists \text{el}$ and $\forall \text{el}$, we require propositional variable $\beta$ to be $i$-bound, i.e. it becomes bound in relation to the world $i$ abbreviated $i \mapsto \beta$.
- In $\square \in$ and the induction rules formula $i \preceq j$ must be the MRA; applying the rule on the step $n$ of the proof, we discard $i \preceq j$ and all subsequent formulas until the step $n$.

Definition 11 (QPTL$_{ND}$ proof) An ND proof of a QPTL formula $B$ is a finite sequence of QPTL$_{ND}$ formulae $A_1, A_2, \ldots, A_n$ which satisfies the following conditions:
reflexivity \[ i \preceq i \]

seriality \[ i \preceq j \quad \text{and} \quad j \preceq i \quad \Rightarrow \quad i \equiv j \]

transitivity \[ \vdash i \preceq j, j \preceq k \quad \Rightarrow \quad i \preceq k \]

\[ \begin{align*}
\text{order}\quad 1 \leq i \leq j \\
\text{seriality}\quad i \preceq j, j \preceq i \quad \Rightarrow \quad i \equiv j \\
\text{transitivity}\quad i \preceq j, j \preceq k \quad \Rightarrow \quad i \preceq k \\
\end{align*} \]

\footnotesize{Figure 3. QPTL\textsubscript{ND}-rules for relational judgements}

\[ \begin{align*}
\Box_{el} & \quad \frac{i : \Box A, i \preceq j}{j : A} \\
\Diamond_{el} & \quad \frac{i : \Diamond A}{i \prec j, j : A} \quad \Rightarrow \quad j \\
\Diamond_{el}^* & \quad \frac{i : \Diamond A}{i \prec j, i' : A} \quad \Rightarrow \quad i' \\
\forall_{el} & \quad \frac{i : \forall \alpha A(\alpha)}{i : A(\alpha/\beta)} \quad \beta \text{ is free for substitution} \\
\exists_{el} & \quad \frac{i : \exists \alpha A(\alpha)}{i : A(\alpha/\beta)} \quad \Rightarrow \quad i \mapsto \beta
\end{align*} \]

\footnotesize{Figure 4. Elimination rules for temporal operators and quantifiers}

- every \( A_i \) \((1 \leq i \leq n)\) is either an assumption, in which case it should have been discarded, or the conclusion of one of the ND rules, applied to some foregoing formulae;
- the last formula, \( A_n \), is \( i : B \), for some label \( i \);
- no variable - world label is flagged twice;
- no variable that has been bound is free in the last formula of the proof or in any formula from Assumptions.

When \( B \) has a QPTL\textsubscript{ND} proof we will abbreviate it as \( \vdash \text{ND} B \).

**Proof Examples.**

Here we provide a number of examples of ND proofs. First, we give an example of the QPTL\textsubscript{ND} proof with detailed comments establishing that the following formula is a theorem.

\[ \exists x \quad [(x \Rightarrow \Box \neg x) \land (\neg \Box x \Rightarrow x)] \quad (2) \]

Though we have not introduced any rigorous search techniques yet, we can still incorporate plenty of intuitive strategies developed on our previous works. Since our main goal is to derive 2, we will first aim at deriving the formula in the scope of \( \exists \), i.e. \( \Box(\neg x \Rightarrow \Box \neg x) \land (\neg \Box x \Rightarrow x) \)), labelled with \( i \). Since the main symbol in this formula is \( \Box \), we would commence the proof by the assumption \( i \preceq j \) and will aim at obtaining \( j : (x \Rightarrow \Box \neg x) \land (\neg \Box x \Rightarrow x) \). Having achieved this we would then have all the grounds to apply the \( \Box_{el} \) rule. Therefore, our new subgoals would be to derive both conjuncts,
Let us now provide several examples of ND proofs including those that show that our side conditions on the rules do not interfere with the derivation of desired formulas. Consider the following formulas:

1. \( j : x \Rightarrow \Box \neg x \) and \( j : \Box \neg x \Rightarrow x \). Steps 2-7 and 8-10 of the proof give these desired derivations. Considering the goal \( j : x \Rightarrow \Box \neg x \) we assume \( j : x \) at step 2 and aim at deriving \( j : \Box \neg x \).

   1. \( i \preccurlyeq j \) \hspace{1cm} \text{assumption}
   2. \( j : x \) \hspace{1cm} \text{assumption}
   3. \( \text{Next}(j, j') \) \hspace{1cm} \Box \text{seriality}
   4. \( j' : \exists x \neg x \) \hspace{1cm} \exists \text{in3}
   5. \( j' : \neg x \) \hspace{1cm} \exists \text{el} 4, j' \Rightarrow x
   6. \( j : \Box \neg x \) \hspace{1cm} \exists \text{in} 3, 5
   7. \( j : x \Rightarrow \Box \neg x \) \hspace{1cm} \Rightarrow \text{in} 6, [2 - 6]

   We derive \( j' : \neg x \) at step 5, introducing two formulae at steps 3 and 4 based on the \( \Box \) seriality and \( \exists \text{in3} \) rule (one of our 0-premise rules) and the elimination of \( \exists \). The latter rule requires to bind variable \( x \) in the world \( j' \). Once we reached \( j' : \neg x \) at step 5, we can reach the goal \( j : \Box \neg x \) at step 6 by introducing \( \Box \) to 3 and 5. Now we apply \( \Rightarrow \text{in} \) rule obtaining 7, thus, deriving the left conjunct. Note that at this stage Assumptions \( = i \preccurlyeq j, j : x \) and we discharge the MRA (\( j : x \)) and discard all formulae [2-6]. Now our aim is to derive the right conjunct \( j : \Box \neg x \Rightarrow x \). Thus, we assume \( j : \Box \neg x \) (step 8) aiming at obtaining the consequent. The latter of course would require to reach \( j' : \neg x \) which is done at step 10 with the intermediate assisting step 9 (another 0-premise rule). Note that step 10, another \( \exists \text{el} \) rule would require to bind \( x \), but now in a different world, \( j \). At step 11 we apply \( \Rightarrow \text{in} \) discharging the MRA (at step 8) and discarding formulae 8-10. At step 12 we apply \( \land \text{in} \) to 7 and 11, and at step 13 we introduce \( \Box \) discharging the MRA (at step 1) and discarding formulae [1-12]. Finally, we derive the desired formula by introducing \( \exists \) to 13.

   8. \( j : \Box \neg x \) \hspace{1cm} \text{assumption}
   9. \( j : \exists x x \) \hspace{1cm} \exists \text{in2}
   10. \( j : x \) \hspace{1cm} \exists \text{el} 9, j \Rightarrow x
   11. \( j : \Box \neg x \Rightarrow x \) \hspace{1cm} \Rightarrow \text{in} 10, [8 - 10]
   12. \( j : (x \Rightarrow \Box \neg x) \land (\Box \neg x \Rightarrow x) \) \hspace{1cm} \land \text{in} 7, 11
   13. \( i : \Box((x \Rightarrow \Box \neg x) \land (\Box \neg x \Rightarrow x)) \) \hspace{1cm} \Box \text{in} 1, 12
   14. \( i : \exists x \Box((x \Rightarrow \Box \neg x) \land (\Box \neg x \Rightarrow x)) \) \hspace{1cm} \exists \text{in} 13

Let us now provide several examples of ND proofs including those that show that our side conditions on the rules do not interfere with the derivation of desired formulas. Consider the following formulas:

\[
\begin{array}{c}
\text{in} \quad j : A, [i, i', x] \quad i : \Box A \\
\Diamond \text{in} \quad i : A \quad \Box \text{in} \quad j' : A, \text{Next}(i, i') \\
\forall \text{ in} \quad i : A(\alpha / \beta) \quad i : \forall A(\alpha)
\end{array}
\]
correspond to the semantics, and, to the reflexivity property of the \(\Box\) and \(\Diamond\) operations.

Proof examples.

\[
\vdash \neg \Diamond A \Rightarrow \Box \neg A
\]  \hspace{1cm} (4)

1. \(i : \neg \Diamond A\) \hspace{1cm} assumption
2. \(i \prec j\) \hspace{1cm} assumption
3. \(i \preceq j\) \hspace{1cm} \(2, \prec - \preceq\) rule
4. \(j : A\) \hspace{1cm} assumption
5. \(i : \Diamond A\) \hspace{1cm} 3, 4, \(\Diamond m\)
6. \(j : \neg A\) \hspace{1cm} 1, 5, \(\neg m, [1 - 5]\),
7. \(i : \Box \neg A\) \hspace{1cm} 5, \(\Box m, [2 - 6]\), \(\Rightarrow\ y\)
8. \(i : \neg \Diamond A \Rightarrow \Box A\) \hspace{1cm} 7, \(\Rightarrow m, [1 - 7]\)

Hence, the following rule is derivable

\[
i : \neg \Diamond A
\frac{}{i : \Box \neg A}
\]  \hspace{1cm} (5)

\[
\vdash \neg \Diamond \neg A \Rightarrow \Box A
\]  \hspace{1cm} (6)

1. \(i : \neg \Diamond \neg A\) \hspace{1cm} assumption
2. \(i \prec j\) \hspace{1cm} assumption
3. \(i \preceq j\) \hspace{1cm} 3, \(\prec - \preceq\) rule
4. \(j : \neg A\) \hspace{1cm} assumption
5. \(i : \Diamond \neg A\) \hspace{1cm} 2, 4, \(\Diamond m\)
6. \(j : \neg A\) \hspace{1cm} 1, 5, \(\neg m, [4 - 5]\),
7. \(j : A\) \hspace{1cm} 6, \(\neg e\)
8. \(i : \Box A\) \hspace{1cm} 7, \(\Box m, [2 - 7]\), \(\Rightarrow\ y\)
9. \(i : \neg \Diamond \neg A \Rightarrow \Box A\) \hspace{1cm} 6, \(\Rightarrow m, [1 - 8]\)

Hence, the following rule is derivable

\[
i : \neg \Diamond \neg A
\frac{}{i : \Box \neg A}
\]  \hspace{1cm} (7)

\[
\vdash \neg \Box A \Rightarrow \Diamond \neg A
\]  \hspace{1cm} (8)

1. \(i : \neg \Box A\) \hspace{1cm} assumption
2. \(i : \neg \Diamond \neg A\) \hspace{1cm} assumption
3. \(i : \Box A\) \hspace{1cm} 2, rule (7)
4. \(i : \neg \Diamond \neg A\) \hspace{1cm} 1, 3, [2 - 3]
5. \(i : \Diamond \neg A\) \hspace{1cm} 4, \(\neg e\)
6. \(i : \neg \Box A \Rightarrow \Diamond \neg A\) \hspace{1cm} 6, \(\Rightarrow m, [1 - 6]\)

Hence, the following rule is derivable

\[
i : \neg \Box A
\frac{}{i : \Diamond \neg A}
\]  \hspace{1cm} (9)

\[
\vdash \neg \Diamond \neg A \Rightarrow \Box \neg A
\]  \hspace{1cm} (10)

1. \(i : \neg \Diamond \neg A\) \hspace{1cm} assumption
2. \(\text{Next}(x, x')\) \hspace{1cm} Next seriality
3. \(i' : A\) \hspace{1cm} assumption
4. \(i : \Box A\) \hspace{1cm} 2, 3, \(\Box m\)
5. \(i' : \neg A\) \hspace{1cm} \(\neg m, [3 - 4]\)
6. \(i : \Box \neg A\) \hspace{1cm} 5, \(\Box m, [2 - 5]\)
7. \(i : \neg \Diamond \neg A \Rightarrow \Box \neg A\) \hspace{1cm} 6, \(\Rightarrow m, [1 - 6]\)
Hence, the following rule is derivable

\[
\frac{i : \neg \Box A}{i : \Box \neg A}\]

(11)

\[\vdash \neg (A \lor B) \Rightarrow (\neg A \land \neg B)\]

(12)

1. \(i : \Box \neg (A \lor B)\) \hspace{1cm} assumption
2. \(i : A\) \hspace{1cm} assumption
3. \(i : A \lor B\) \hspace{1cm} \lor_{in_1}, 2
4. \(i : \neg A\) \hspace{1cm} \neg_{in}, 1, 3, [2 \rightarrow 3]
5. \(i : B\) \hspace{1cm} assumption
6. \(i : A \lor B\) \hspace{1cm} \lor_{in_2}, 5
7. \(i : \neg A\) \hspace{1cm} \neg_{in}, 1, 6, [5 \rightarrow 6]
8. \(i : \neg A \land \neg B\) \hspace{1cm} \land_{in}, 4, 7
9. \(i : \neg (A \lor B) \Rightarrow (\neg A \land \neg B)\) \hspace{1cm} \Rightarrow_{in}, 8, [1 \rightarrow 8]

Hence, the following rule is derivable

\[
\frac{i : \neg (A \lor B)}{i : \neg A \land \neg B}\]

(13)

\[\vdash \Box (A \lor B) \Rightarrow (\Box A \lor \Box A)\]

(14)

1. \(i : \Box (A \lor B)\) \hspace{1cm} assumption
2. \(i : \neg (\Box A \lor \Box B)\) \hspace{1cm} assumption
3. \(i : \neg \Box A \land \neg \Box B\) \hspace{1cm} 2, rule (13)
4. \(i : \neg \Box A\) \hspace{1cm} 3, \land_{el}
5. \(i : \neg \Box A \Rightarrow \Box \neg A\) \hspace{1cm} (11)
6. \(i : \Box \neg A\) \hspace{1cm} 4, 5, \Rightarrow_{el}
7. \(i : \neg \Box B\) \hspace{1cm} 3, \land_{el}
8. \(i : \neg \Box B \Rightarrow \Box \neg B\) \hspace{1cm} (11)
9. \(i : \Box \neg B\) \hspace{1cm} 7, 8, \Rightarrow_{el}
10. \text{Next}(x, x') \hspace{1cm} 1, \Box_{el}, \mapsto x'
11. \(i' : A \lor B\) \hspace{1cm} 1, \Box_{el}
12. \(i' : \neg A\) \hspace{1cm} 6, \Box_{el}
13. \(i' : \neg B\) \hspace{1cm} 9, \Box_{el}
14. \(i' : B\) \hspace{1cm} 11, 12, \lor_{el}
15. \(i : \neg (\Box A \lor \Box A)\) \hspace{1cm} \neg_{in}, 13, 14, [2 \rightarrow 14]
16. \(i : \Box A \lor \Box A\) \hspace{1cm} \neg_{el}, 15
17. \(i : \Box (A \lor B) \Rightarrow (\Box A \lor \Box A)\) \hspace{1cm} 17, \Rightarrow_{in}, [1 \rightarrow 16]

Now we show that our improved side conditions on the rules do correspond to the semantics, namely, to the reflexivity property of the \(\Box\) and \(\Diamond\) operations. Here we would like to show that

\[
\Diamond x = x \lor \Box \Diamond x
\]

(15)

and

\[
\Box x = x \land \Box \Box x
\]

(16)

Below we will present proofs for both directions in (15).
Proof for $x \lor \Diamond x \Rightarrow \Diamond x$

1. $i : x \lor \Diamond x$ \hspace{1cm} \text{assumption}
2. $i : \neg \Diamond x$ \hspace{1cm} \text{assumption}
3. $i : \Box \neg x$ \hspace{1cm} 2, rule\(5\)
4. $i \preceq i$ \hspace{1cm} \preceq \text{reflexivity}
5. $i : \neg x$ \hspace{1cm} \preceq \text{reflexivity}
6. $i : \Diamond x$ \hspace{1cm} \preceq \text{reflexivity}
7. $i' : \Diamond x$ \hspace{1cm} \preceq \text{reflexivity}
8. $i' \prec j$ \hspace{1cm} \preceq \text{reflexivity}
9. $j : x$ \hspace{1cm} \preceq \text{reflexivity}
10. $i \preceq i'$ \hspace{1cm} \preceq \text{reflexivity}
11. $i \prec j$ \hspace{1cm} \preceq \text{reflexivity}
12. $j : \neg x$ \hspace{1cm} \preceq \text{reflexivity}
13. $i : \neg \neg \Diamond x$ \hspace{1cm} \preceq \text{reflexivity}
14. $i : \Diamond x$ \hspace{1cm} \preceq \text{reflexivity}
15. $i : x \lor \Diamond x$ \hspace{1cm} \preceq \text{reflexivity}

For the proof for $\Diamond x \Rightarrow x \lor \Diamond x$ let us first remind the method to deal with the linearity \([4]\)

$$\Gamma, Lin(i, j) \vdash \Delta, \bot \quad \Rightarrow \quad \Gamma, Lin(i, j), i \preceq j \vdash \Delta, \bot$$

where $Lin(i, j)$ abbreviates the linearity constraint $(i \preceq j) \lor (i \simeq j) \lor (j \preceq i)$. This searching rule represents reasoning by cases. If a linearity constraint is introduced into the proof and the current goal is $\bot$ then the rule requires to derive $\bot$ making each of the disjuncts of the linearity constraints in turn as a new assumption.

1. $i : \Diamond x$ \hspace{1cm} \text{assumption}
2. $i : \neg(x \lor \Diamond x)$ \hspace{1cm} \text{assumption}
3. $i \prec j$ \hspace{1cm} $\Diamond x$\(1\)
4. $j : x$ \hspace{1cm} $\Diamond x$\(1\), $\rightarrow$ $j$
5. $i : \neg x \land \neg \Diamond x$ \hspace{1cm} 2, rule\(12\)
6. $i : \neg x$ \hspace{1cm} 5, $\land$\(\text{el}\)
7. $i : \neg \Diamond x$ \hspace{1cm} 5, $\land$\(\text{el}\)
8. $i : \Box \neg x$ \hspace{1cm} \preceq \text{reflexivity}
9. $i' : \neg x$ \hspace{1cm} 8, $\land$\(\text{el}\),$\rightarrow$ $i'$
10. $i' : \Box \neg x$ \hspace{1cm} 9, rule\(5\)
11. $\text{Next}(i, i')$ \hspace{1cm} $\Box$ \text{reflexivity}
12. $i \preceq i'$ \hspace{1cm} $\Box$ \text{reflexivity}
13. $i' \preceq j \lor i' \simeq j \lor j \preceq i'$ \hspace{1cm} \simeq \text{linearity}
14. $i' \preceq j$ \hspace{1cm} case studies, 13
15. $j : \neg x$ \hspace{1cm} $\Box x$, 12, 14
16. $i' \simeq j$ \hspace{1cm} case studies, 13
17. $j : \Box \neg x$ \hspace{1cm} 5, 11
18. $j \preceq j$ \hspace{1cm} reflexivity
19. $j : \neg x$ \hspace{1cm} $\Box x$, 12, 18
20. $j \preceq i'$ \hspace{1cm} case studies, 13
21. $j \simeq i'$ \hspace{1cm} 3, 20
22. $j : \neg x$ \hspace{1cm} $\Box x$, 12, 21
23. $\bot$ \hspace{1cm} 13, 4, 15, 19, 22, \text{Reasoning by cases}
24. $i : \neg \neg(x \lor \Diamond x)$ \hspace{1cm} $\neg x$, 2, 23, \([2 - 23]\)
25. $i : x \lor \Diamond x$ \hspace{1cm} $\neg x$, 24
26. $i : \Diamond x \Rightarrow x \lor \Diamond x$ \hspace{1cm} $\Rightarrow x$, 25, \([1 - 25]\)

A few more proof examples will be given in the next section.
4 Correctness

In this section we will establish meta-theoretical properties of the QPTL\textsubscript{ND} system defined above. Namely, we will show that QPTL\textsubscript{ND} is sound (§4.1) and complete (§4.2).

4.1 Soundness

Lemma 2 Let $\Gamma = \{C_1, C_2, \ldots, C_k\}$ be a set of QPTL formulae such that $\mathbf{\hat{\Gamma}} = \{C_1, C_2, \ldots, C_k\}$, where each $C_i (1 \leq i \leq k)$ is $j : C_i$, for some label $j$, is a set of non-discarded assumptions which are contained in the QPTL\textsubscript{ND} proof for a QPTL formula $B$, at some step, $m$. Let $\Lambda$ be a set of QPTL\textsubscript{ND} formulae in the proof at step $m$ such that for any $D, \mathbf{D} \in \Lambda$ if it is obtained by an application of some ND rule, and let $\Delta$ be a conclusion of a QPTL\textsubscript{ND} rule which is applied at step $m + 1$. Let $\mathbf{\hat{\Gamma}}^*$ consist of all assumptions from $\mathbf{\hat{\Gamma}}$ that have not been discharged by the application of this rule, the same for a set $\Lambda^*$. Then if $\mathbf{\hat{\Gamma}}^*$ is realisable in a model $\sigma$ then $\Lambda^* \cup \Delta$ is also realisable in $\sigma$.

Proof: We prove this lemma by induction on the number of QPTL\textsubscript{ND} rules applied in the proof. Thus, assuming that lemma is correct for the number, $n$, of the QPTL\textsubscript{ND} rules, we must show that it is also correct for the $n + 1$-th rule.

The proof is quite obvious for the rules for Booleans. We only show the most interesting case where the rule of $\forall \mathbf{\hat{\Gamma}}$ is applied.

Case $\forall \mathbf{\hat{\Gamma}}$. Let $i : A$ be an element of $\mathbf{\hat{\Gamma}}$ which is the most recent non-discharged assumption in the proof. An application of the rule $\forall \mathbf{\hat{\Gamma}}$, at step $m + 1$, gives a QPTL\textsubscript{ND} formula $i : \neg A$ as a conclusion. This means that at some earlier steps of the proof we have $j : C$ and $j : \neg C$. Here we should consider several subcases that depend on the set to which these contradictory QPTL\textsubscript{ND} formulae belong. We now prove the lemma for some of these cases. Subcase 1. Assume that both $j : C$ and $j : \neg C$ are in the set $\mathbf{\hat{\Gamma}}^*$ but nor $j : \neg C$ coincides with $i : A$. Then the statement that the realisability of $\mathbf{\hat{\Gamma}}^*$ implies the realisability of $\Lambda^* \cup \{i : \neg A\}$ is true simply because $\mathbf{\hat{\Gamma}}^*$ is not realisable. Subcase 2. Assume that both $j : C$ and $j : \neg C$ are in the set $\Lambda$. Then if the set $\mathbf{\hat{\Gamma}}$ is realisable, the set $\Lambda$ should be realisable as well. But, as assumed, it is not. So, $\mathbf{\hat{\Gamma}}$ also can not be realisable. Note that $\mathbf{\hat{\Gamma}} = \mathbf{\hat{\Gamma}}^* \cup \{i : A\}$. It should be clear that if $\mathbf{\hat{\Gamma}}$ is realisable then also $\{i : \neg A\}$ is. If we think of the set $\mathbf{\hat{\Gamma}}$ as an initial part of the proof, then the set $\Lambda$ is empty after the deletion of the corresponding steps of proof. In this case we are done.

Cases with the rules for temporal operators that do not require restrictions on labels can be shown straightforwardly from the semantics. Let us consider cases with the rules that require restrictions, for example, the rule $\diamondsuit \mathbf{\hat{\Gamma}}$. Case $\diamondsuit \mathbf{\hat{\Gamma}}$. Let $i : \diamondsuit A \in \Lambda$. We have to show realisability of $\Lambda^* \cup \{i : A\}$ provided that realisability of $\mathbf{\hat{\Gamma}}^*$ holds. Actually $\mathbf{\hat{\Gamma}}^* = \mathbf{\hat{\Gamma}}$ and $\Lambda^* = \Lambda$ in this case. By induction hypothesis we know that realisability of $\mathbf{\hat{\Gamma}}$ implies realisability of $\Lambda$. Then for a mapping $f(i) = s_i$, we have $\langle \sigma, s_i \rangle \models \diamondsuit A$. From the latter, according to the semantics, it follows that there exists a state $s_k$, $i \leq k$, such that $\langle \sigma, s_k \rangle \models A$. Now we can define a mapping $f'$, the extension of $f$, as $f' = f \cup \{(j, s_k)\}$. This mapping is correct because the variable $j$ was not used in the proof before, otherwise it should be flagged twice. So, the mapping $f$ is not defined for $j$. We can see that $\langle \sigma, f(j) \rangle \models A$ and the pair $(i, j)$ satisfies the criteria of Definition 7. Therefore, $\Lambda^* \cup \{i : A\}$ is realisable.

Case $\exists \mathbf{\hat{\Gamma}}$. Let $i : A(y)$ be an element of a set $\Lambda$. Again, we should prove realisability of $\Lambda^* \cup \{i : \exists x A(x)\}$ assuming that $\mathbf{\hat{\Gamma}}^*$ is realizable. Here we also find $\mathbf{\hat{\Gamma}}^* = \mathbf{\hat{\Gamma}}$ and $\Lambda^* = \Lambda$ since the rule $\exists \mathbf{\hat{\Gamma}}$ does not require elimination of assumptions. According to induction hypothesis realisability of $\mathbf{\hat{\Gamma}}$ implies realisability of $\Lambda$. Let $f(i) = s_k$ and $\langle \sigma, f(i) \rangle \models A(y)$. This means that there is some interpretation of the variable $y$ which provides satisfiability of $A(y)$ in the state $i$ of $\sigma$. Now from the definition 7 we know that $\langle \sigma, f(i) \rangle \models \exists x A(x)$ iff there exists $x$-variant of $\sigma$ which makes $A(x)$ true in $\langle \sigma, i \rangle$. Actually we only know that there exists $y$-variant of $\sigma$ for $A(y)$ to be true at $\langle \sigma, i \rangle$. If $y$ is free for substitution with respect to $x$, then $A(y)$ is a substitution variant of $A(x)$ and $y$-variant of $\sigma$ makes the same effect as $x$-variant of $\sigma$. So, $\Lambda \cup \{i : \exists x A(x)\}$ is realisable.

Cases $\exists \mathbf{\hat{\Gamma}}_1$, $\exists \mathbf{\hat{\Gamma}}_3$. These 0-premise rules are easy to justify just because the statements of the kind $\exists x x$ and $\exists x \neg x$ are always true — in each state of every model. To see this, examine an arbitrary model $\sigma$ and arbitrary state $s_i \in \sigma$. If $x \not\in s_i$, consider an $x$-variant of $\sigma$ such that $x \in s_i$ — interpretation of other variables is irrelevant. The same reasoning works for $\exists x \neg x$.

Case $\exists \mathbf{\hat{\Gamma}}_2$. This case is justified by the model theoretical arguments in the same manner as presented in the previous cases. But it is important to avoid the situation when after eliminating an existential quantifier we asseret that some propositional variable $y$ is true but probably asserted $y$ to be false before by application of the same rule $\exists \mathbf{\hat{\Gamma}}_2$. It is easy to derive this kind of unforced contradiction from $\exists y y$ and $\exists y \neg y$. The flagging tool prevents us from this kind of collision. (END)
Theorem 1 [QPTL$_{ND}$ Soundness] Let $A_1, A_2, \ldots, A_k$ be a QPTL$_{ND}$ proof of QPTL formula $B$ and let $\hat{\Gamma} = \{C_1, C_2, \ldots, C_n\}$ be a set of QPTL formulae such that $\hat{\Gamma} = \{C_i, C_2, \ldots, C_n\}$, where each $C_i$ ($1 \leq i \leq n$) is $j : C_i$, for some label $j$, is a set of discharged assumptions which occur in the proof. Then $\models_{ND} B$, i.e. $B$ is a valid formula.

PROOF: Consider the proof $A_1, A_2, \ldots, A_k$ for some QPTL formula $B$. According to Definition 11, $A_k$ has the form $i : B$, for some label $i$. In general, $i : B$ belongs to some set, $\Lambda$, of non-discarded QPTL$_{ND}$ formulae in the proof. By Lemma 2 we can conclude that realisability of $\hat{\Gamma}$ implies realisability of $\Lambda$. But $\hat{\Gamma}$ is empty and, therefore, is realisable in any model and for any function $f$ by Definition 7. So $\Lambda$ is also realisable in any model and for any function $f$. That is, any formula that belongs to $\Lambda$ is valid. In particular $i : B$ is valid. (END)

4.2 Completeness

We will prove the completeness of QPTL$_{ND}$ by showing that every theorem of the following axiomatics for QPTL [12] is a theorem of QPTL$_{ND}$.

Axioms for QPTL (schemes).

A0. Schemes for classical propositional logic
A1. $\square (A \Rightarrow B) \Rightarrow (\square A \Rightarrow \square B)$
A2. $\square A \Rightarrow (A \land \circ \square A)$
A3. $\circ \neg A \Rightarrow \neg \circ A$
A4. $\circ (A \Rightarrow B) \Rightarrow (\circ A \Rightarrow \circ B)$
A5. $\square (A \Rightarrow \circ A) \Rightarrow (A \Rightarrow \square A)$
QX0. $\exists x (x \land \circ \neg x)$
QX1. $\forall x \circ A \Rightarrow \circ \forall x A$
QX2. $\forall x A \Rightarrow A(B/x)$, $x$ is free for $B$ in $A$

Rules:

\[
\begin{array}{c}
\text{MP} \\
\hline
\vdash A, \quad \vdash A \Rightarrow B \quad \vdash A \Rightarrow B \\
\hline
\vdash B
\end{array}
\]

\[
\begin{array}{c}
\text{Gen} \\
\hline
\vdash A \quad \vdash A \Rightarrow B \\
\hline
\vdash \square A \quad \vdash A \Rightarrow \forall x B
\end{array}
\]

In \forall Gen rule $x \notin Var(A)$.

In the following Induction rule let $X = \{x_0, x_1, \ldots, x_m\}$ and $Y = \{y_0, x_1, \ldots, y_m\}$ and $X \cap Y = \emptyset$. A formula $B$ has a temporal depth $k$ in $X$ if every $x \in X$ does not occur in the scope of $\circ$ or more than $k$ $\circ$ operators. The induction rule is formulated below where $\beta$ has temporal depth $1$ in $X$ and variables of $Y$ do not appear in $\beta$.

\[
\begin{array}{c}
\vdash B \Rightarrow \exists y_0 \ldots \exists y_n \circ \leq^1 (\bigwedge_{i=0}^n (x_i \equiv y_i) \land \circ B[Y/X]) \\
\hline
\vdash B \Rightarrow \exists x \square B
\end{array}
\]

To prove the completeness of QPTL$_{ND}$ we first show that every instance of the scheme of the above axiomatics is a theorem of QPTL$_{ND}$, and, secondly, that given that the assumptions of the rules of the axiomatics have a QPTL$_{ND}$ proof then so do their conclusions.

Lemma 3 Every instance of the scheme of the QPTL axiomatics is a theorem of QPTL$_{ND}$.

PROOF: Since QPTL$_{ND}$ extends the natural deduction system for classical propositional logic, all classical schemes are provable in QPTL$_{ND}$ by a simple modification of classical proofs introducing a world label, say $i$, for any formula of a classical proof [2].

Since QPTL$_{ND}$ extends PLTL$_{ND}$, all schemes A0-A5, for the axiomatics for PLTL, are provable in QPTL$_{ND}$, following the completeness of PLTL$_{ND}$.

Now we will present proofs for some of the QPTL schemes from the set of axioms above.

Proof for an instance of Axiom QX0. $QX0 = \exists x (x \land \circ \square \neg x)$
Proof for an instance of Axiom QX1.

1. $i : \exists x x$ assumption
2. $\exists_{in2}$
3. $\exists_{seriality}$
4. $i' \leq j$ assumption
5. $j : \exists x \neg x$ $\exists_{in3}$
6. $i' : \Box \neg x$ $\Box_{in3}, 5, [3 - 5]$  
7. $i : \Box \neg x$ $\Box_{in2}, 6$
8. $i : x$ $\exists_{in1}, 9$
9. $i : x \land \Box \neg x$ $\land_{in7}, 8$
10. $i : \exists x (x \land \Box \neg x)$ $\exists_{in1}$

Proof for an instance of Axiom QX2.

1. $i : \forall x \Box A(x)$ assumption
2. $i : \Box A(x)$ $\forall_{el}$
3. $Next(i, i')$ $\Box_{seriality}$
4. $i' : A(x)$ $\Box_{el}$
5. $i' : \forall x A(x)$ $\forall_{in4}, x i' - marked$
6. $i : \Box \forall x A(x)$ $\Box_{in5}$
7. $i : \forall x (A(x) \Rightarrow \Box \forall x A(x))$ $\Rightarrow_{in6}$

It is easy to establish the following proposition.

**Proposition 1** Let $A_1, A_2, \ldots, A_n$ be a $QPTL_{ND}$ proof of a $QPTL$ formula $B$. Let $B'$ be obtained from $B$ by substituting a subformula $C$ of $B$ by $C'$. Then $A'_1, A'_2, \ldots, A'_n$, where any occurrence of $C$ is substituted by $C'$, is a $QPTL_{ND}$ proof of $B'$.

Hence by Proposition 1 and the proofs of the instances of QPTL axioms we obtain the proof for Lemma 3.

(End)

**Lemma 4** If $A$ has a $QPTL_{ND}$ proof then $\Box A$ also has a $QPTL_{ND}$ proof.

**Proof:** Consider some arbitrarily chosen theorem of $QPTL_{ND}$, $A$, and let $x$ and $y$ be the world variables that do not occur in this proof.

Now we start a new proof commencing it with the assumption that $\neg \Box A$ (below, to make the proof more transparent, we will scorn the rigorous presentation of $QPTL_{ND}$ proof, writing metaformulae instead of the QPTL formulae which can be justified based upon Proposition 1):

1. $i : \neg \Box A$ assumption
2. $i : \Diamond \neg A$ $1, \neg \Box transformation$
3. $i \leq j$ $2, \Diamond_{el}, \mapsto j$
4. $j : \neg A$ $2, \Diamond_{el}$

At this stage we are coming back to the proof of $A$ noticing that at the last step of this proof we have a formula $z : A$ (recall that $k \neq i \neq j$). In this proof of $A$ we do the following: change every occurrence of $k$ to $j$. Obviously, we still have a proof for $A$. Take this newly generated proof (say it has $n$ steps) and write it continuing steps 1-4 of the proof above. Thus, we
Lemma 5 If $A \Rightarrow B$ and $A$ have QPTL\textsubscript{ND} proofs then $B$ also has an QPTL\textsubscript{ND} proof.

PROOF: Let the proofs for $A \Rightarrow B$ and $A$ have $n$ and $m$ steps respectively. Since both are QPTL\textsubscript{ND} proofs we can rewrite these proofs such that they would have completely different sets of the world labels. Let the last formula of the proof for $A \Rightarrow B$ be $i : A \Rightarrow B$. We commence constructing the QPTL\textsubscript{ND} proof for $B$ as follows:

1. \hspace{1em} (first formula of the proof for $A \Rightarrow B$)

\[ n \text{. } i : A \Rightarrow B \quad (last \text{ formula of the proof for } A \Rightarrow B) \]

Now we can change to $x$ the label that occurs at the last step of the proof for $A$ and continue the construction of the proof for $B$ as follows.

\[ n + 1 \text{. } i : A \Rightarrow B \quad (last \text{ formula of the proof for } A \Rightarrow B) \]

It is easy to establish that this proof, by its construction, satisfies the criteria for the proof. (END)

Lemma 6 If $A \Rightarrow B$ has QPTL\textsubscript{ND} proofs then $A \Rightarrow \forall x.B$ (where $x \not\in Var(A)$) also has an QPTL\textsubscript{ND} proof.
Consider some arbitrarily chosen theorem of QPTL\textsubscript{ND}, \( A \Rightarrow B \), and let \( i \) be the world variable that do not occur in this proof.

Now we start a new proof commencing it with the assumption that \( i : \neg(A \Rightarrow \forall x.\, B) \) (below, to make the proof more transparent, we will scorn the rigorous presentation of QPTL\textsubscript{ND} proof, writing metaformulae instead of the QPTL formulae which can be justified based upon Proposition 1):

\[
\begin{align*}
1. & \ i : \neg(A \Rightarrow \forall x.\, B) \quad \text{assumption} \\
2. & \ i : A \land \exists x.\, \neg B \quad 1, \neg \Rightarrow, \land \forall \text{ transformation} \\
3. & \ i : A \quad 2, \exists el, \Rightarrow x \\
4. & \ \text{(first formula of the proof for } A \Rightarrow B) \\
5. & \ \text{...} \\
6. & \ \text{...} \\
7. & \ i : A \Rightarrow B \quad (last formula of the proof for } A \Rightarrow B) \\
8. & \ i : A \land \exists x.\, \neg B \quad \land el, 2 \\
9. & \ i : A \land \neg B \quad \land el, 2 \\
10. & \ i : B \quad \Rightarrow el, n + 4, n + 5 \\
11. & \ i : \neg(A \Rightarrow \forall x.\, B) \quad \neg el, n + 6, n + 7, 1 - (n + 8) \\
12. & \ i : A \Rightarrow \forall x.\, B \quad \neg el, n + 8 \\
\end{align*}
\]

(\text{END})

The following lemma is important when we manage a proof for derivability of propositional induction rule. Let \( \Gamma \) be a set of QPTL formulae \( \{C_1, \ldots, C_n\} \) such that \( \Gamma = \{C_1, C_2, \ldots, C_k\} \), where each \( C_i \) \((1 \leq i \leq k)\) is \( j : C_i \), for some label \( j \), is a set of non-discarded assumptions which are contained in the QPTL\textsubscript{ND} proof for a QPTL formula \( B \), at some step, \( m \). Let \( k : D(X) \) is a QPTL\textsubscript{ND} formula of step \( m \) with all variables from the set \( X \) unbounded. Then we have the following lemma.

**Lemma 7**

\[ \Gamma, k : D(X), k \land \exists C \equiv Y \vdash_{QPTL_N D} k : \exists D(Y/X) \]

implies

\[ \Gamma, k : D(X), k \land \exists C \equiv Y \vdash_{QPTL_N D} k : D(X) \]

for all \( m \).

**PROOF:** The proof proceeds by induction on the structure of QPTL formula \( D(X) \). For the sake of simplicity we assume that sets of propositional variables \( X \) and \( Y \) are singletons. The generalization to the arbitrary finite sets of variables is a routine task. The basis of induction is quite simple. Consider some derivation of \( k : \exists C \equiv y \) from \( \Gamma \) and \( k : \exists C \equiv y \land \neg y \). It is easy to see that \( k : \exists C \equiv y \) is also derivable from the same premisses by \( \exists el, \land el, \Rightarrow el, \land ser. \) and \( \land in \). Now we should examine the cases when \( D \) is a complex formula obtained by an application of corresponding introduction rule. Let us explore some of them. Let \( D(x) \) is of the form \( E \land F \) and \( k' : D(y) \) is obtained by \( \land in \) from \( k' : E \) and \( k' : F \) (where one of these formulae or both contain \( y \) ). By induction hypothesis we have two statements: \( \Gamma', k : \exists C \equiv y \vdash_{QPTL_N D} k : \exists C(x) \) and \( \Gamma', k : \exists C \equiv y \vdash_{QPTL_N D} k : \exists C(x) \). Let \( T_1 \) and \( T_2 \) be the derivations of \( k : \exists C \) and \( k : \exists C \) respectively. We can concatenate \( T_1 \) and \( T_2 \) to obtain a new derivation which ends with \( k' : E \land F \) (introduced by \( \land in \)). So, at the next stage we have \( k : \exists C \land F \). Next suppose \( D(x) \) is a formula with external propositional quantifier, for instance \( \exists z C(z) \). Note that \( z \neq y \). We can obtain the result straightforwardly by induction hypothesis and \( \exists in \). The other cases are treated in the similar manner. (END)
Lemma 8 Assume we have a QPTL\textsubscript{ND} proof for a formula $A \Rightarrow \exists X(\bigcirc (X \Rightarrow Y \land Y \Rightarrow X) \land \bigcirc A(Y/X))$. Then we also have QPTL\textsubscript{ND} proof for $A \Rightarrow \exists X \bigcirc A$. Proof: We exploit the fact that a formula $\alpha = A \Rightarrow \exists X(\bigcirc (X \Rightarrow Y \land Y \Rightarrow X) \land \bigcirc A(Y/X))$ is provable in QPTL\textsubscript{ND}. Let $T$ be an example of such a proof. Choose labels $i, j$ which have no occurrences in $T$. Now assume that our ND derivation starts as follows:

1. $i : A$ assumption
2. $i \geq j$ assumption

Next step is to insert properly the proof $T$ into our derivation above. Assume that the last formula of $T$ is $k : A \Rightarrow \exists X(\bigcirc (X \Rightarrow Y \land Y \Rightarrow X) \land \bigcirc A(Y/X))$. Note that $k \neq i \neq j$. Next rewrite all occurrences of $k$ in $T$ with $j$. So, we have:

1. $i : A(X)$ assumption
2. $i \geq j$ assumption
3. $\ldots$ (first formula of the proof of $\alpha$)

$\ldots$

$n, j : \alpha (last\ formula\ of\ the\ proof\ of\ \alpha)$
$n + 1, j : A(X)$ assumption
$n + 2, j : \exists X(\bigcirc (X \Rightarrow Y \land Y \Rightarrow X) \land \bigcirc A(Y/X)) \Rightarrow_{el} n, n + 1$
$n + 3, j : \bigcirc (X \Rightarrow Y \land Y \Rightarrow X) \land \bigcirc A(Y/X) \Rightarrow_{el} \exists el \ n + 3$
$n + 4, j : \bigcirc (X \Rightarrow Y \land Y \Rightarrow X) \land \bigcirc A(Y/X) \Rightarrow_{el} \exists el \ n + 3$

Now we apply lemma 7 to obtain the next step of the reasoning.

$n + 6, j : \bigcirc A(X)$
$n + 7, j : A(X) \Rightarrow \bigcirc A(X) \Rightarrow_{im} n + 6 \ [n + 1 + 6]$
$n + 8, j : \bigcirc A(X) \Rightarrow_{im} n + 6 \ [n + 1 + 6]$

$\ldots$

$n + m, \exists X \bigcirc A(X)$
$n + m + 1, A(X) \Rightarrow \exists X \bigcirc A(X) \Rightarrow_{im} n + m \ [1 - n + m]$

(END)

Now we are ready to prove the completeness of QPTL\textsubscript{ND}.

Theorem 2 [QPTL\textsubscript{ND} Completeness] For any QPTL\textsubscript{ND} formula, $A$, if $\models_{\text{ND}} A$ then there exists a QPTL\textsubscript{ND} proof of $A$.

Proof: Consider an arbitrarily chosen theorem, $A$, of QPTL. By induction on $n$, the length of the axiomatic proof for $A$, we now show that $A$ also has a QPTL\textsubscript{ND} proof.

Base Case. $n = 1$. In this case $A$ is one of the schemes of the QPTL axiomatics, and thus, the base case follows from Lemma 3.

Induction step. If Theorem 2 is correct for the proof of the length $m$, $(1 \leq m \leq n)$ then it is correct for the proof of the length $m + 1$.

Here the formula at the step $m + 1$ is either an axiom or is obtained from some previous formulae either by generalisation or the modus ponens rules. The proof for these cases follows from Lemma 4 and Lemma 6 respectively.

Therefore, given that $A$ has an axiomatic proof it also has a QPTL\textsubscript{ND} proof.

(END)

5 Discussion

We have presented a natural deduction system for propositional linear time temporal logic and established its correctness. To the best of our knowledge, there is only one other ND construction, in [17] for the full QPTL which is based upon the developments in [21]. In Marchigoni’s construction, many rules such as $\lor_{el}, \Rightarrow_{im}, \neg_{in}$ and rules for $U$, are formulated in so called ‘indirect’ fashion, i.e. they allow us to transform some given proofs. From our point of view, these rules are much more complex than the rules in our system, and thus would be more difficult for developing a proof-searching procedure.
Although a proof-searching technique for this novel construction is still an open, and far from being trivial, problem, we expect to incorporate many of the methods previously defined for classical propositional and first-order logics.

The study of complexity of the method for both classical and temporal framework, in turn, is another component of future research as well as the extension of the approach to capture the branching-time framework.

6 Discussion

We have presented a natural deduction system for the propositional linear time temporal logic extended with propositional quantifiers and established its correctness. To the best of our knowledge, there are no other ND constructions for this logic. Additionally, we have simplified the technical side of the approach, requiring now fewer side conditions for the rules and clarifying some notions accompanying the proof.

The presented construction enables us to reason in a natural fashion in this new expressive framework.

The study of complexity of the method for both classical and temporal framework, in turn, is another component of future research as well as the extension of the approach to capture the branching-time framework.

Note that our aim was to establish a natural deduction setting which would be sufficient to capture QPTL. As part of future work we plan, on the one hand, to refine the set of rules to simplify proofs and on the other hand, to work towards more efficient proof search procedures.

In this paper we also did not mention anything specific about the complexity issues. It is known that QPTL has a non-elementary complexity [16] and since our system has been shown to be deductively equivalent to the axiomatics for QPTL, we know the upper bound of our system. But the complexity study would be more interesting being applied to the proof search, which as we mentioned is our next project.

In fact it is a wider project than dealing with the QPTL logic only. It would work with the classical logic first, investigating various modifications of the set of rules and it effects on the proof search and its complexity. Secondly, it would tackle the rules for temporal operators, and first of all, the problems arising with the induction. Due to the construction of the system similar to our previous work on the ND setting for the first order logic, we have sufficient grounds that we can adapt the basic ideas and main principles of the proof search used for the predicate logic, but this, of course, is a separate project and we also expect to incorporate many of the methods previously defined for classical propositional and first-order logics.

References


