A Simpler formulation of natural deduction calculus for linear-time temporal logic

Bolotov, A., Grigoriev, O. and Shangin, V.

A conference paper published in: Bhanu, P. (ed.) The Proceedings of the 3rd Indian International Conference on Artificial Intelligence, Pune, India, December 17-19, 2007. IICAI. pp. 1253-1266.

A Simpler Formulation of Natural Deduction Calculus for Linear-Time Temporal Logic

Alexander Bolotov*, Oleg Grigoriev** and Vasilyi Shangin**

* Harrow School of Computer Science
University of Westminster
Watford Road, Harrow HA1 3TP, UK.
A.Bolotov@wmin.ac.uk
http://www2.wmin.ac.uk/bolotoa/index.html
** Department of Logic, Faculty of Philosophy, Moscow State University, Moscow, 119899, Russia.
{shangin,grig}@philos.msu.ru

Abstract. The paper continues our studies of natural deduction calculus for the propositional linear-time temporal logic PLTL. We present a new formulation of natural deduction calculus for PLTL. The system is shown to be sound and complete. This new formulation is simpler than the previous one, and this fact is believed to be crucial for possible applications of our technique as an automatic reasoning tool in a deliberative decision making framework across various AI applications.

1 Introduction

The paper is a sequel to the authors' studies of natural deduction calculus for the propositional linear-time temporal logic PLTL [6]. In the previous paper, we presented a natural deduction proof system for PLTL and established its correctness [1]. Being based on the techniques for a variety of classical and non-classical logics [1–3], that system served as a background for a natural deduction system for computation temporal logic CTL [4], in turn.

In this paper, we present a new formulation of natural deduction calculus accompanied with the updated correctness argument. A new system is simpler than the previous one, and this fact is believed to be crucial for possible applications of our technique as an automatic reasoning tool in a deliberative decision making framework across various AI applications.

In a new formulation of the system two rules for Until operator elimination together with the induction rule are replaced with only one rule for Until operator elimination. Moreover, instead of three rules for Until operator introduction, we now present only one simple introduction rule. This effect appears as a result of making use of a new rule, which is an analog of the least fixpoint axiom in the standart axiomatization of the system PLTL.

A simpler formulation of the system allows us to make the correctness argument more transparent. Therefore, we update the correctness argument (especially, the soundness argument). We believe that a new formulation of the system will improve the efficiency of the proof searching procedures.

The paper is organized as follows. In §2 we review the syntax and semantics of PLTL in §2.1. In §2.2 we describe the ND for PLTL henceforth referred to as $\mathrm{PLTL_{ND}}$ and give an example of the construction of the proof. Subsequently, in §3, we provide the correctness argument. Finally, in §4, we provide concluding remarks and identify future work.

2 Natural Deduction System $\mathbf{PLTL_{ND}}$

In this section we review the logic PLTL and the calculus $\mathrm{PLTL_{ND}}$.

2.1 Syntax and Semantics of PLTL

In the syntax of PLTL we identify a set, *Prop*, of atomic propositions:

$$p, q, r, \dots, p_1, q_1, r_1, \dots, p_n, q_n, r_n, \dots$$

classical operators: $\neg, \wedge, \Rightarrow, \vee$, and temporal operators: $\Box$ ('always in the future'), $\Diamond$ ('at sometime in the future'), $\bigcirc$ ('at the next moment in time'), and $\mathcal{U}$ ('until').

The set of *well-formed formulae* of PLTL, wff_{PLTL} is defined as follows.

Definition 1 (PLTL syntax).

1. *All atomic propositions (members of Prop) are in* wff_{PLTL}.
2. *If* A *and* B *are in* wff_{PLTL}, *then so are* $A \wedge B$, $\neg A$, $A \vee B$, *and* $A \Rightarrow B$.
3. *If* A *and* B *are in* wff_{PLTL}, *then so are* $\Box A$, $\Diamond A$, $\bigcirc A$, *and* $A \mathcal{U} B$.

For the semantics of PLTL we utilise the notation of [5]. A model for PLTL formulae, is a discrete, linear sequence of states $\sigma = s_0, s_1, s_2, \dots$ which is isomorphic to the natural numbers, $\mathcal{N}$, and where each state, s_i, $0 \leq i$, consists of the propositions that are true in it at the i-th moment of time. If a well-formed formula A is satisfied in the model σ at the moment i then we abbreviate it by $\langle \sigma, i \rangle \models A$. Below, in Figure 1, we define the relation $\models$, where indices $i, j, k \in \mathcal{N}$.

Definition 2 (PLTL Satisfiability). *A well-formed formula, A, is satisfiable if, and only if, there exists a model* σ *such that* $\langle \sigma, 0 \rangle \models A$.

Definition 3 (PLTL Validity). *A well-formed formula, A, is valid if, and only if, A is satisfied in every possible model, i.e. for each* σ, $\langle \sigma, 0 \rangle \models A$.

2.2 The Calculus $\mathbf{PLTL_{ND}}$

Here we present the formulation of $\mathrm{PLTL_{ND}}$ with a slightly different set of rules in comparison with its original formulation in [1]. Namely, now we have new rules, application of negation to $\mathcal{U}$ and $\Diamond$ operators, and the rule representing one of the De Morgan laws, but fewer rules for $\mathcal{U}$ (see details below).

$\langle \sigma, i \rangle \models p$	iff $p \in s_i$, for $p \in Prop$	
$\langle \sigma, i \rangle \models \neg A$	iff $\langle \sigma, i \rangle \not\models A$	
$\langle \sigma, i \rangle \models A \wedge B$	iff $\langle \sigma, i \rangle \models A$ and $\langle \sigma, i \rangle \models B$	
$\langle \sigma, i \rangle \models A \vee B$	iff $\langle \sigma, i \rangle \models A$ or $\langle \sigma, i \rangle \models B$	
$\langle \sigma, i \rangle \models A \Rightarrow B$	iff $\langle \sigma, i \rangle \not\models A$ or $\langle \sigma, i \rangle \models B$	
$\langle \sigma, i \rangle \models \Box A$	iff for each j if $i \leq j$ then $\langle \sigma, j \rangle \models A$	
$\langle \sigma, i \rangle \models \Diamond A$	iff there exists j such that $i \leq j$ and $\langle \sigma, j \rangle \models A$	
$\langle \sigma, i \rangle \models \bigcirc A$	iff $\langle \sigma, i+1 \rangle \models A$	
$\langle \sigma, i \rangle \models A \,\mathcal{U}\, B$	iff there exists j such that $i \leq j$ and $\langle \sigma, j \rangle \models B$ and for each k, if $i \leq k < j$ then $\langle \sigma, k \rangle \models A$	

Fig. 1. Semantics for PLTL

The core idea of a natural deduction proof technique for a logic L is to establish rules of the following two classes: *elimination* rules which decompose formulae and *introduction* rules aimed at constructing formulae, introducing new logical constants. Given a task to prove some formula A of L, we aim at synthesising A. Every proof commences with an assumption and, in general, we are allowed to introduce assumptions at any step of the proof. In the type of natural deduction that we are interested in, assumptions have conditional interpretation. Namely, given that a formula A is preceded in a proof by assumptions $C_1, C_2, \dots C_n$ we interpret this situation as follows: if $C_1, C_2, \dots C_n$ are satisfiable in L then A is satisfiable in L. Thus, if A is a theorem (a valid formula in L) and we want to obtain its proof then we must interpret A 'unconditionally', i.e. it should not depend on any assumptions. In our system, the corresponding process is called *discarding* of assumptions, which accompanies the application of several introduction rules. As we will see below, in a proof of a theorem in our system the set of non-discarded assumptions should be empty.

Another feature of our construction of PLTL_{ND} is the use of the labeling technique. In the language of PLTL_{ND} we use labeled PLTL formulae and a specific type of expressions that use labels themselves, called *relational judgements*. Thus, additionally to elimination and introduction rules, we also establish rules to manipulate with relational judgements.

Extended PLTL Syntax and Semantics.

We extend the PLTL language by introducing labels. Labels are terms, elements of the set, $Lab = \{x, y, z, x_1, x_2, x_3, \dots\}$, where $x, y, z \dots$ are variables. When constructing a PLTL_{ND} proof, we associate formulae appearing in the proof with a model σ described in §2.1 such that labels in the proof are interpreted over the states of σ. Since σ is isomorphic to natural numbers, we can introduce the operations on labels: $\simeq$, which stands for the equality between labels, $\preceq$ and $\prec$, which are syntactic analogues of the $\leq$ and $<$ relation in σ. Thus, $\preceq$ satisfies the following properties:

(2.1) For any $i \in Lab:\ i \preceq i$ (reflexivity),

(2.2) For any $i, j, k \in Lab$ if $i \preceq j$ and $j \preceq k$ then $i \preceq k$ (transitivity).
(2.3) For any $i, j, k \in Lab$ if $i \preceq j$ and $i \preceq k$ then $j \prec k$ or $k \prec j$ or $j \simeq k$ (linearity).
(2.4) For any $i \in Lab$, there exists $j \in Lab$ such that $i \preceq j$ (seriality).

Now, we define a relation $Next \subset Lab^2 : Next(x, y) \Leftrightarrow x \prec y$ and there is no $z \in Lab$ such that $x \prec z$ and $z \prec y$.

$Next$ is the 'predecessor-successor' relation which satisfies the seriality property: for any $i \in Lab$, there exists $j \in Lab$ such that $Next(i, j)$.

Let $'$ abbreviate the operation which being applied to $i \in Lab$ gives us $i' \in Lab$ such that $Next(i, i')$.

As we have already mentioned above, now we are able to introduce the expressions representing the properties of relations '$\preceq$', '$\prec$', '$\simeq$' and 'Next', and the operation $'$ which, following [7], we call *relational judgements*.

Definition 4 (PLTL$_{\text{ND}}$ Syntax).

- *If A is a PLTL formula and $i \in Lab$ then $i : A$ is a* PLTL$_{\text{ND}}$ *formula.*
- *Any relational judgement of the type $Next(i, j)$, $i \preceq j$, $i \prec j$ and $i \simeq j$ is a* PLTL$_{\text{ND}}$ *formula.*

Some useful and rather straightforward properties relating operations on labels are given below.

(2.5) For any $i, j \in Lab$ if $Next(i, j)$ then $i \preceq j$.
(2.6) For any $i, j \in Lab$ if $i \prec j$ then $i \preceq j$.

For the interpretation of PLTL$_{\text{ND}}$ formulae we adapt the semantical constructions defined in §2.1 for the logic PLTL. In the rest of the paper we will use capital letters $A, B, C, D, \ldots$ as metasymbols for PLTL formulae, and calligraphic letters $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D} \ldots$ to abbreviate formulae of PLTL$_{\text{ND}}$, i.e. either labelled formulae or relational judgements. The intuitive meaning of $i : A$ is that A is satisfied at the world i.

Let Γ be a set of PLTL$_{\text{ND}}$ formulae, let $D_\Gamma = \{x | x : A \in \Gamma\}$, let σ be a model as defined in §2.1 and let f be a function which maps elements of D_Γ into $\mathcal{N}$ (recall that a PLTL model σ is isomorphic to natural numbers).

Definition 5 (Realisation of PLTL$_{\text{ND}}$ formulae in a model). *Model σ* realises *a set, Γ, under a mapping, $f^\sigma : Lab \longrightarrow \sigma$, if the following conditions hold:*

(1) For any $x \in Lab$, and for any PLTL$_{\text{ND}}$ *formula A, if $x : A \in \Gamma$ then $\langle \sigma, f(x) \rangle \models A$,*
(2) For any x, y, if $x \preceq y \in \Gamma$, and $f^\sigma(x) = i$, and $f(y) = j$ then $i \leq j$,
(3) For any x, y, if $Next(x, y) \in \Gamma$, and $f^\sigma(x) = i$, and $f^\sigma(y) = j$ then $j = i+1$.

The set Γ in this case is called realisable *in σ under a mapping f^σ. If a model and a mapping are clear from the context, the set Γ is simply called realisable.*

Definition 6 (PLTL$_{\mathbf{ND}}$ Logical Consequence). *A set of* PLTL$_{\text{ND}}$ *formulae* Γ *logically implies a* PLTL$_{\text{ND}}$ *formula* $\mathcal{A}$, *denoted* $\Gamma \models_{ND} \mathcal{A}$, *if*

1. *All elements of* Γ *and* $\mathcal{A}$ *are of the form* $i : C$ *(for some PLTL formula* C*) and prefixed with the same label,* i*,*
2. *a* PLTL$_{\text{ND}}$ *formula* $\mathcal{A}$ *is realisable whenever a set* Γ *is, for each mapping* f^σ *and every model* σ*.*

Definition 7 (PLTL$_{\mathbf{ND}}$ Validity). *A well-formed* PLTL$_{\text{ND}}$ *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*.*

Rules of Natural Deduction System.

In Figure 2 we define these sets of elimination and introduction rules, where prefixes '*el*' and '*in*' abbreviate an elimination and an introduction rule, respectively.

Elimination Rules :	Introduction Rules :
$\wedge\ el_1 \quad \dfrac{i:A \wedge B}{i:A}$	$\wedge\ in \quad \dfrac{i:A, \quad i:B}{i:A \wedge B}$
$\wedge\ el_2 \quad \dfrac{i:A \wedge B}{i:B}$	$\vee\ in_1 \quad \dfrac{i:A}{i:A \vee B}$
$\vee\ el \quad \dfrac{i:A \vee B, \quad i:\neg A}{i:B}$	$\vee\ in_2 \quad \dfrac{i:B}{i:A \vee B}$
$\Rightarrow el \quad \dfrac{i:A \Rightarrow B, \quad i:A}{i:B}$	$\Rightarrow\ in \quad \dfrac{[i:C], \quad i:B}{i:C \Rightarrow B}$
$\neg\ el \quad \dfrac{i:\neg\neg A}{i:A}$	$\neg\ in \quad \dfrac{[j:C], \quad i:B, \quad i:\neg B}{j:\neg C}$

Fig. 2. PLTL$_{\mathbf{ND}}$-rules for Booleans

– In the formulation of the rules '$\Rightarrow$ *in*' and '$\neg$ *in*' formulae $[i:C]$ and $[j:C]$ respectively must be the most recent non discarded [3] assumption occurring in the proof. When we apply one of these rules on step n and discard an assumption on step m, we also discard all formulae from m to $n-1$. We will write $[m$ - $(n-1)]$ to indicate this situation.

We keep the notions of *flagged* and *relatively flagged* label with the meaning similar to the notions of flagged and relatively flagged variable in first order logic [3]. By saying that the label, j, is flagged, abbreviated as $\mapsto j$, we mean that it is bound to a state and, hence, cannot be rebound to some other state. By saying that a variable i is relatively flagged (bound) by j, abbreviated as $j \mapsto i$ we mean that a bounded variable, j, restricts the set of runs for i that is linked to it in the relational judgment, for example $i \preceq j$.

Now in Figure 3 we introduce the following rules to manipulate with relational judgements which correspond to the properties (2.1)-(2.6).

reflexivity $\dfrac{}{i \preceq i}$ $\qquad$ $\bigcirc$ *seriality* $\dfrac{}{Next(i, i')}$

$\prec / \preceq$ $\dfrac{i \prec j}{i \preceq j}$ $\qquad$ $\bigcirc / \preceq$ $\dfrac{Next(i, i')}{i \preceq i'}$

transitivity $\dfrac{i \preceq j,\ j \preceq k}{i \preceq k}$

$\preceq$ *linearity* $\dfrac{i \preceq j,\ i \preceq k}{(j \preceq k) \vee (j \simeq k) \vee (k \preceq j)}$

Fig. 3. PLTL$_{\mathbf{ND}}$-rules for relational judgements

The linearity rule needs some additional comments. Strictly speaking, in the PLTL$_{\mathrm{ND}}$ 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.

Next, in Figure 4 we define elimination and introduction rules for the temporal logic operators.

$\star$ When applying $\bigcirc_{el}$ the conclusion $i' : A$ becomes marked by M_1. This affects other rules:

- the condition $\forall C(j\!:\!C \notin M1)$ in the rule $\Diamond_{el}$ means that the label j should not occur in the proof in any formula, $j\!:\!C$, that is marked by $M1$,

- the condition $j\!:\!A \notin M1$ in the rule $\Box_{in}$ means that $j\!:\!A$ is not marked by $M1$.

$\star\star$ In $\Box_{in}$ formula $i \preceq j$ must be the most recent assumption and a variable j is *new* in a derivation. Applying the rule on the step n of the proof, we discard $i \preceq j$ and all subsequent formulae until the step n.

Finally, we add the following three rules:

$$\neg\mathcal{U}\ \dfrac{i\!:\!\neg(A\,\mathcal{U}\,B)}{i\!:\,\Box\neg B \vee \neg B\,\mathcal{U}\,(\neg A \wedge \neg B)} \qquad \neg\Diamond\ \dfrac{i\!:\!\neg\Diamond A}{i\!:\,\Box\neg A} \qquad \neg\vee\ \dfrac{i\!:\!\neg(A \vee B)}{i\!:\!\neg A \wedge \neg B}$$

Elimination Rules :

$$\Box el \quad \frac{i:\Box A, \quad i \preceq j}{j : A}$$

$$\Diamond el \quad \frac{i:\Diamond A}{i \preceq j, \quad j : A} \qquad \begin{matrix}\forall C(j:C \notin M1) \\ \mapsto j,\ j \mapsto i\end{matrix}$$

$$\bigcirc el^{\star} \quad \frac{i:\bigcirc A}{i':A} \qquad i':A \in M1$$

$$\mathcal{U}_{el} \quad \frac{i:\Box(B \Rightarrow C), i:\Box((A \wedge \bigcirc C) \Rightarrow C)}{i:A\mathcal{U}B \Rightarrow C}$$

Introduction Rules :

$$\Box in^{\star\star} \quad \frac{j:A, \quad [i \preceq j]}{i:\Box A} \qquad \begin{matrix} j:A \notin M1 \\ \mapsto j,\ j \mapsto i\end{matrix}$$

$$\Diamond in \quad \frac{j:A, \quad i \preceq j}{i:\Diamond A}$$

$$\bigcirc in \quad \frac{i':A, \quad Next(i,i')}{i:\bigcirc A}$$

$$\mathcal{U}\, in \quad \frac{i:B}{i:A\mathcal{U}B}$$

Fig. 4. Temporal ND-rules

The third rule, $\neg\vee$, simply represents one of De Morgan laws and is derivable from the set of classical rules mentioned above. The rule $\neg\mathcal{U}$ is not derivable from the set of rules for temporal operators given above. Their addition, together with the use of fewer rules for $\mathcal{U}$, leads us to a new ND formulation of PLTL.

Definition 8 ($\mathbf{PLTL_{ND}}$ Derivation). *A derivation $\mathfrak{D}$ of a* $\mathrm{PLTL_{ND}}$ *formula $\mathcal{A}$ from a set of* $\mathrm{PLTL_{ND}}$ *formulae Γ, which are called premises, is a nonempty sequence of* $\mathrm{PLTL_{ND}}$ *formulae such that*

1. *All elements of Γ and $\mathcal{A}$ are of the form $i : C$ (for some PLTL formula C and $i \in Lab$) and prefixed with the same label, i,*
2. *each member of $\mathfrak{D}$ is either an element of a set Γ, or an assumption, or a result of an application of one of the derivation rules of* $\mathrm{PLTL_{ND}}$ *system,*
3. *none of labels occurring in $\mathfrak{D}$ is flagged twice or flagges itself,*
4. *a label of $\mathcal{A}$ is not flagged in $\mathfrak{D}$,*
5. *a set of nondiscarded assumptions is empty.*

An expression $\Gamma \vdash_{ND} \mathcal{A}$ denotes the fact that there is a $\mathrm{PLTL_{ND}}$ *derivation of $\mathcal{A}$ from Γ.*

Definition 9 ($\mathbf{PLTL_{ND}}$ Proof). *An ND proof of a PLTL formula B is a finite sequence of* $\mathrm{PLTL_{ND}}$ *formulae $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ which satisfies the following conditions:*

- *every $\mathcal{A}_i$ $(1 \leq i \leq n)$ 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, $\mathcal{A}_n$, is $x:B$, for some label x,*

– *no variable - world label is flagged twice or relatively binds itself.*

When B has a PLTL_{ND} proof we will abbreviate it as $\vdash_{ND} B$ indicating that B is a theorem.

Examples

Here we present the proof of the $\Box$ induction principle and some examples of the reasoning based on the application of the natural deduction system introduced above that are needed for the proof of induction.

Firstly, it is easy to justify the generalisation principle [1], and show that contraposition and transitivity of $\Rightarrow$ rules are derivable:

$$If\ \vdash_{ND} A\ then\ \vdash_{ND} \Box A. \tag{1}$$

$$\frac{A \Rightarrow B}{\neg B \Rightarrow \neg A} \tag{2}$$

$$\frac{A \Rightarrow B, B \Rightarrow C}{A \Rightarrow C} \tag{3}$$

The following theorem is also easily (classically) proved:

$$\vdash_{ND} (\mathbf{true} \wedge A) \Rightarrow A \tag{4}$$

Next we present proofs for PLTL theorems that characterise some properties of $\bigcirc$, $\Diamond$, $\Box$ and $\mathcal{U}$ operators:

$$\vdash_{ND} \bigcirc\neg A \Rightarrow \neg\bigcirc A \tag{5}$$

1. $x : \bigcirc\neg A$	*assumption*
2. $x' : \neg A$	$\bigcirc_{el}, 1$
3. $x : \neg\neg\bigcirc A$	*assumption*
4. $x : \bigcirc A$	$3,\ \neg_{el}$
5. $x' : A$	$4,\ \bigcirc_{el}$
6. $x : \neg\neg\neg\bigcirc A$	$\neg_{in}, 2, 5, [3-5]$
7. $x : \neg\bigcirc A$	$\neg_{el}, 6$
8. $x : \bigcirc\neg A \Rightarrow \neg\bigcirc A$	$\Rightarrow_{in}, 7, [1-7]$

$$\vdash_{ND} \neg\Diamond A \Rightarrow \Box\neg A \tag{6}$$

1. $x : \neg\Diamond A$	*assumption*
2. $x \preceq y$	*assumption*
3. $y : A$	*assumption*
4. $x : \Diamond A$	$2, 3, \Diamond_{in}$
5. $y : \neg A$	$1, 4, \neg_{in}, [3-4]$
6. $x : \Box\neg A$	$5, \Box_{in}, [2-5],\ \ \mapsto y,\ y \mapsto x$
7. $x : \neg\Diamond A \Rightarrow \Box\neg A$	$6, \Rightarrow_{in}, [1-6]$

$$\vdash_{ND} \neg\Box A \Rightarrow \Diamond\neg A \tag{7}$$

$$\begin{array}{ll}
1.\ x : \neg \Box A & \textit{assumption} \\
2.\ x : \neg \Diamond \neg A & \textit{assumption} \\
3.\ x : \neg \Diamond \neg A \Rightarrow \Box A & \textit{theorem}\ (6) \\
4.\ x : \Box A & 2, 3, \Rightarrow el \\
5.\ x : \neg\neg \Diamond \neg A & 1, 4,\ \neg_{in}, [2-4] \\
6.\ x : \Diamond \neg A & 5,\ \neg_{el} \\
7.\ x : \neg \Box A \Rightarrow \Diamond \neg A & 6,\ \Rightarrow_{in}, [1-6]
\end{array}$$

$$\vdash_{ND} A \mathcal{U}\, \mathbf{false} \Rightarrow \mathbf{false} \tag{8}$$

$$\begin{array}{ll}
1.\ x : A \wedge \bigcirc \mathbf{false} \Rightarrow \mathbf{false} & \textit{classical theorem} \\
2.\ x : \Box(A \wedge \bigcirc \mathbf{false} \Rightarrow \mathbf{false}) & 1,\ \textit{rule}\ (1) \\
3.\ x : \mathbf{false} \Rightarrow \mathbf{false} & \textit{classical theorem} \\
4.\ x : \Box(\mathbf{false} \Rightarrow \mathbf{false}) & 3,\ \textit{rule}\ (1) \\
5.\ x : A \mathcal{U}\, \mathbf{false} \Rightarrow \mathbf{false} & 2, 4,\ \mathcal{U}_{el}
\end{array}$$

$$\vdash_{ND} \Diamond A \Rightarrow (\mathbf{true}\, \mathcal{U}\, A) \tag{9}$$

$$\begin{array}{ll}
1.\ x : \Diamond A & \textit{assumption} \\
2.\ x \preceq y & 1, \Diamond_{el}, \mapsto y, y \mapsto x \\
3.\ y : A & 1, \Diamond_{el} \\
4.\ x : \neg(\mathbf{true}\, \mathcal{U}\, A) & \textit{assumption} \\
5.\ x : \Box \neg A \vee \neg A \mathcal{U}\, (\neg A \wedge \mathbf{false})) & 4,\ \neg \mathcal{U} \\
6.\ x : \Box \neg A \vee \neg A \mathcal{U}\, \mathbf{false} & 5,\ \textit{classical} \\
7.\ x : \Box \neg A & 6,\ \textit{rule}\ 8,\ \textit{classical} \\
8.\ y : \neg A & 7, 2,\ \Box_{el} \\
9.\ x : \neg\neg(\mathbf{true}\, \mathcal{U}\, A) & 3, 8,\ \neg_{in},\ [4-8] \\
10.\ x : \mathbf{true}\, \mathcal{U}\, A & 9,\ \neg_{el} \\
11.\ x : \Diamond A \Rightarrow (\mathbf{true}\, \mathcal{U}\, A) & 10, \Rightarrow_{in}, [1-10]
\end{array}$$

$$\vdash_{ND} \bigcirc \Diamond A \Rightarrow \Diamond A \tag{10}$$

$$\begin{array}{ll}
1.\ x : \bigcirc \Diamond A & \textit{assumption} \\
2.\ Next(x, x') & \bigcirc\ \textit{seriality} \\
3.\ x' : \Diamond A & 1, \bigcirc_{el} \\
4.\ x \preceq x' & 2, Next/\preceq \\
5.\ x' \preceq z & 3, \Diamond_{el},\ \mapsto z,\ z \mapsto x' \\
6.\ z : A & 3, \Diamond_{el} \\
7.\ x \preceq z & 4, 5, \preceq\ \textit{transitivity} \\
8.\ x : \Diamond A & \Diamond_{in}, 6, 7 \\
9.\ x : \bigcirc \Diamond A \Rightarrow \Diamond A & \Rightarrow_{in}, [1-9]
\end{array}$$

From theorem (10) we can easily derive

$$\vdash_{ND} (\mathbf{true} \wedge \bigcirc \Diamond A) \Rightarrow \Diamond A \tag{11}$$

From theorem (11) by generalisation rule (1) we have

$$\vdash_{ND} \Box((\mathbf{true} \wedge \bigcirc \Diamond A) \Rightarrow \Diamond A) \tag{12}$$

Now we are ready to prove the $\Box$ induction:

$$\vdash_{ND} (\Box(A \Rightarrow \bigcirc A) \wedge A) \Rightarrow \Box A \tag{13}$$

1. $x : \Box(A \Rightarrow \bigcirc A) \wedge A$	assumption
2. $x : \Box(A \Rightarrow \bigcirc A)$	$1, \wedge_{el}$
3. $x : A$	$1, \wedge_{el}$
4. $x : \neg \Box A$	assumption
5. $x : \neg \Box A \Rightarrow \Diamond \neg A$	*theorem* (7)
6. $x : \Diamond \neg A$	$4, 5, \Rightarrow_{el}$
7. $x : \Diamond \neg A \Rightarrow (\mathbf{true}\, \mathcal{U} \neg A)$	*theorem* (9)
8. $x : (\mathbf{true}\, \mathcal{U} \neg A)$	$6, 7, \Rightarrow_{el}$
9. $x : \Box(\neg A \Rightarrow \neg A)$	*rule* (1) *applied to classical theorem*
10. $x \preceq v$	assumption
11. $v : A \Rightarrow \bigcirc A$	$2, 10, \Box_{el}$
12. $v : \neg \bigcirc A \Rightarrow \neg A$	11, *rule* (2)
13. $v : \bigcirc \neg A \Rightarrow \neg \bigcirc A$	*theorem* (5)
14. $v : \bigcirc \neg A \Rightarrow \neg A$	12, 13, *rule* (3)
15. $v : (\mathbf{true} \wedge \bigcirc \neg A) \Rightarrow \bigcirc \neg A$	*theorem* (4)
16. $v : (\mathbf{true} \wedge \bigcirc \neg A) \Rightarrow \neg A$	14, 15, *rule* (3)
17. $x : \Box((\mathbf{true} \wedge \bigcirc \neg A) \Rightarrow \neg A)$	$\Box_{in}, 10, 16, [10-16], \mapsto v,\ v \mapsto x$
18. $x : (\mathbf{true}\, \mathcal{U} \neg A) \Rightarrow \neg A$	$9, 17, \mathcal{U}_{el}$
19. $x : \neg A$	$8, 18, \Rightarrow_{el}$
20. $x : \neg\neg \Box A$	$\neg_{in}, 3, 19, [4-19]$
21. $x : \Box A$	$\neg_{el}, 20$
22. $x : (\Box(A \Rightarrow \bigcirc A) \wedge A) \Rightarrow \Box A$	$21, \Rightarrow_{in}, [1-21]$

3 PLTL$_{\mathrm{ND}}$ Correctness

3.1 PLTL$_{\mathrm{ND}}$ Soundness

Now let us turn to the proof of a soundness theorem for a system of PLTL$_{\mathrm{ND}}$. But before we should prove an important lemma which is substantial for the main theorem. The formulation of this lemma involve a number of details and may seem a bit sophisticated. But the main idea here is quite clear. We want to be convinced that if a set of given premises and non-discarded assumptions is realisable, i.e. has a model in a sense, then a set of all other formulae in a derivation (which are obtained by applications of the rules) is also realisable. Then a proof of soundness theorem follows from this result almost directly.

Lemma 1. *Assume that the following conditions are given:*

- *Let* $\mathfrak{D}$ *be a derivation of some* PLTL$_{\mathrm{ND}}$ *formula* $\mathcal{B}$ *from a set of* PLTL$_{\mathrm{ND}}$ *formulae* Γ,
- $\Phi_m \subseteq \mathfrak{D}$ *is a union of* Γ *and a set of non-discarded assumptions* Θ_m *which are contained in* $\mathfrak{D}$ *at some step* m.

- Λ_m *is a set of* PLTL$_{\rm ND}$ *formulae of* $\mathfrak{D}$ *at the step* m *such that for any* $\mathcal{B}$*, if* $\mathcal{B} \in \Lambda_m$*, then it is obtained by an application of some derivation rule, and let* Δ *be a conclusion of a* PLTL$_{\rm ND}$ *rule which is applied at step* $m+1$.
- Φ_{m+1} *consists of the set* Γ *and all assumptions from* Θ_m *that have not been discarded by the application of this rule;* Λ_{m+1} *consists of the non-discarded members of* Λ_m *and the elements of a set* Δ.

Then, for all f^σ *and* σ*, if* Φ_{m+1} *is realisable in a model* σ *under* f^σ *then* Λ_{m+1} *is also realisable in* σ *under* f^σ.

Proof. We prove this lemma by induction on the number of PLTL$_{\rm ND}$ rules applied in the derivation. Thus, assuming that lemma is correct for the number, n ($n \in \mathbb{N}$), of the applications of the PLTL$_{\rm ND}$ rules, we must show that it is also correct for $n+1$.

Case $\Rightarrow_{in}$. Suppose that $x : B$ is some PLTL$_{\rm ND}$ formula in the derivation and $x : A$ is the most recent assumption contained in the set Φ_m. An application of the rule $\Rightarrow_{in}$ results in a PLTL$_{\rm ND}$ formula $x : A \Rightarrow B$ in $\mathfrak{D}$. To prove the lemma for this case, we should consider several subcases depending on the place in the proof where $x : B$ is positioned or the whole structure of the proof.
Subcase 1. Let $x : B \in \Lambda_m$ and an initial part of $\mathfrak{D}$ consists of the elements of Γ directly followed by all the elements of Θ_m. Let us refer to this configuration by expression that Φ_m is an *initial part* of the derivation. After application of the rule $\Rightarrow_{in}$ we have $x : A \Rightarrow B$ on $m+1$-th step of the proof and also the sets $\Phi_{m+1} = \Phi_m - \{x : A\}$ (because $x : A$ is the most recent assumption), $\Lambda_{m+1} = \Delta = \{x : A \Rightarrow B\}$ (because all steps starting from the most recent assumption until the result of the application of the rule are discarded). Thus it should be shown that the set $\{x : A \Rightarrow B\}$ is realisable under each f^σ in every model σ provided that realisability of Φ_{m+1} is assumed. Note that if some model realises Φ_{m+1} but rejects formula A, then this model realises the set $\{x : A \Rightarrow B\}$ by the truth condition for implication. So, assume that for some model both Φ_{m+1} and $\{x : A\}$ are realisable. Recall that the union of these sets is the set Φ_m. By induction hypothesis we know that realisability of Φ_m implies realisability of Λ. But $x : B \in \Lambda$, hence $\{x : A \Rightarrow B\}$ is realisable.
Subcase 2. In this case we consider the situation when $x : B \in \Lambda$ but some elements of Θ_m appeared in the proof after a number of applications of PLTL$_{\rm ND}$ rules have been made. This time the set Λ_{m+1} may contain some elements apart from $x : A \Rightarrow B$. The difficult part is concerned with the case when some model, say σ', realises the set Φ_{m+1}, under some f'^σ, but not realises $\{x : A\}$. As before, we know that this model realises the set $\{x : A \Rightarrow B\}$ but the realisability of the rest of Λ_{m+1} is in question. Let us denote $\Lambda_{m+1} - \{x : A \Rightarrow B\}$ as Λ^*_{m+1}. Now we should appeal to the structure of the proof. Let Φ^p_{m+1} be a subset of Φ_{m+1} consisting of all assumptions which precede the formulae of Λ^*_{m+1} in the proof. As we know, some applications of the rules were made after all of the elements of Λ^*_{m+1} have appeared in the proof. So, by induction hypothesis we have that realisability of Φ^p_{m+1} implies realisability of Λ^*_{m+1}. Suppose that Λ^*_{m+1} is not realisable in σ' under a mapping f'^σ. Then the set Φ^p_{m+1} is also not realisable in this model. But $\Phi^p_{m+1} \subseteq \Phi_{m+1}$ and Φ_{m+1} is realisable in σ', which

is a contradiction. The case when Φ_{m+1} and $\{x : A\}$ are both realisable in σ' follows from the induction hypothesis as was shown in the previous subcase.
Subcase 3. Assume that $x : B \in \Theta_m$. In this case $\Phi_{m+1} = \Phi_m - \{x : A\}$, $\Lambda_{m+1} = \Lambda^*_{m+1} \cup \{x : A \Rightarrow B\}$, where $\Lambda^*_{m+1} \subseteq \Lambda_m$. In this case an argument similar to the previous subcases should work.

In the subsequent part of the proof we will not specially consider the situation as in the subcase 2 above, because the proof for the situation of this kind uses essentially the same routine and then the reasoning similar to that one in subcase 1. So, henceforth we restrict ourselves to the assumption that Φ_m is an initial part of the derivation.

Case $\neg_{in}$. Let $x : A$ be an element of Φ_m and the most recent non-discarded assumption in the proof. An application of the rule $\neg_{in}$ at step $m + 1$ gives a PLTL$_{\text{ND}}$ formula $x : \neg A$ as a conclusion. This means that at some earlier steps of the proof we have $y : C$ and $y : \neg C$. Here we should consider several subcases that depend on which sets these contradictory PLTL$_{\text{ND}}$ formulae belong to. We now prove the lemma for some of these cases.
Subcase 1. Both $y : C$ and $y : \neg C$ are in Φ_m but nor $y : C$ neither $y : \neg C$ coincides with $x : A$. After an application of the rule $\neg_{in}$ we have $\Phi_{m+1} = \Phi_m - \{x : A\}$. Then the statement that the realisation of Φ_{m+1} implies the realisation of Λ_{m+1} is true simply because Φ_{m+1} is not realisable.
Subcase 2. Assume that both $y : C$ and $y : \neg C$ are in the set Λ_m. Then, by induction hypothesis, if the set Φ_m realisable, the set Λ_m should be realisable as well. But, as assumed, Λ_m is not realisable. Therefore, Φ_m also can not be realisable. Note that $\Phi_m = \Phi_{m+1} \cup \{x : A\}$. So, if we suppose that Φ_{m+1} is realisable in some model σ under some f^σ then the set $\{x : A\}$ is not realisable. Hence, $\{x : \neg A\}$ is realisable.
Subcase 3. One of the contradictory PLTL$_{\text{ND}}$ formulae, say $y : C$, belongs to the set Φ_m, while another to Λ_m. Assume that Φ_m is realisable in some model σ under some mapping f^σ. Applying the induction hypothesis, we conclude that Λ_m should be realisable in σ under f^σ. But then realisable the union of these sets, which contains the contradictory elements. So, Φ_m is not realisable. Now we should prove that a realisation of Φ_{m+1} implies a realisation of $\{x : A\}$. Suppose that some σ' and f'^σ provide a realisation of $\Phi m + 1$. So, taking into account that Φ_m cannot be realisable, we can conclude that $\{x : A\}$ is not realisable. This means that $\{x : \neg A\}$ is realisable.

The are some cases to consider when one of the contradictory PLTL$_{\text{ND}}$ formulae coincides with the most recent assumption in the derivation. But these are only slight modifications of the cases shown above.

Case $\Box_{in}$. Suppose that for some PLTL formula A and label, y, $y : A$ is contained in $\mathfrak{D}$ after m-th application of PLTL$_{\text{ND}}$ rule and there is also a relational judgment $x \preceq y \in \Phi_m$, which is the most recent non-discarded assumption. An application of the rule $\Box_{in}$ provides a new PLTL$_{\text{ND}}$ formula, $x : \Box A$, in the derivation. Again, several subcases are required, depending on the place of $x : A$ in $\mathfrak{D}$.
Subcase 1. Φ_m is an initial part of the derivation, $x : A \in \Lambda_m$ and $\Lambda_{m+1} = \Delta =$

$\{x : \Box A\}$. As the induction hypothesis suggests, if the set Φ_m is realisable in a model σ under some f^σ, so then Λ_m is also realisable. Suppose that Φ_{m+1} is realisable in a model, say σ_1, under a mapping f_1^σ. Now let j be an element of σ_1 such that $f_1(x) \leqslant j$. Note that the function f_1^σ is defined for the variable x because of the presence of $x : \Box A$ in Φ_{m+1} but not defined for y as we have deleted from $\mathfrak{D}$ all the formulae containing y. Also note that x and y are necessarily different variables because the situation when $\mapsto x$, $x \mapsto x$ is prohibited in a derivation. Next extend f_1^σ to f_2^σ in a way that $f_2^\sigma = f_1^\sigma \cup \{(y,j)\}$. It is easy to see that Φ_m is realisable under f_2^σ and, by induction hypothesis, Λ_m is also realisable under f_2^σ. But $y : A$ is in Λ_m, so $\langle \sigma, j\rangle \models A$. By virtue of an arbitrary choice of element j we can conclude that $\langle \sigma, f_2^\sigma(x)\rangle \models \Box A$. In view of $f_2(x) = f_1(x)$ we have $\langle \sigma, f_1^\sigma(x)\rangle \models \Box A$ as required.

Subcases 2 and 3 describe a situation which is similar to what presented in the corresponding subcases of the case $\Rightarrow_{in}$ and exploit an analogous reasoning.

Case $\Box_{el}$. For some PLTL formula A and labels x, y PLTL_{ND} formulae $x : A$ and $x \preceq y$ are in the derivation $\mathfrak{D}$. Let us just consider the case when both $x : A$ and $x \preceq y$ are in the set Λ_m. After an application of the rule $\Box_{el}$ we have $\Phi_{m+1} = \Phi_m$, $\Lambda_{m+1} = \Lambda_m \cup \{y : A\}$. By induction hypothesis, realisation of Φ_m for some σ and f^σ implies realisation of Λ_m, which means that $\langle \sigma, f^\sigma(x)\rangle \models \Box A$ and $f^\sigma(x) \leq f^\sigma(y)$. By the truth condition for necessity, we derive that $\langle \sigma, f^\sigma(y)\rangle \models A$. So, Λ_{m+1} is realisable.

Case $\mathcal{U}_{el}$. Both $x : \Box(B \Rightarrow C)$ and $x : \Box((A \wedge \bigcirc C) \Rightarrow C)$ are in Λ_m. An application of the rule $\mathcal{U}_{el}$ does not affect the set Φ_m, so $\Phi_m = \Phi_{m+1}$. Also $\Lambda_{m+1} = \Lambda_m \cup \Delta$. Now assume that Φ_{m+1} is realisable. It follows that $\Lambda_{m+1} - \Delta$ is realisable. It remains to show that Δ is realisable. Let f^σ and σ be corresponding function and model. We have to show that $\langle \sigma, f(x)\rangle \models A \mathcal{U} B \Rightarrow C$. Suppose that $\langle \sigma, f(x)\rangle \models A \mathcal{U} B$. First consider the case when $\langle \sigma, f(x)\rangle \models B$. By induction hypothesis we know that $\langle \sigma, f(x)\rangle \models \Box(B \Rightarrow C)$ (recall that $x : \Box(B \Rightarrow C)$ is in Λ_m), hence $\langle \sigma, f(x)\rangle \models B \Rightarrow C$ and then $\langle \sigma, f(x)\rangle \models C$ as required. Now let $f(x) < j$ and $\langle \sigma, j\rangle \models B$ and for all i such that $f(x) \leqslant i < j$ we have $\langle \sigma, i\rangle \models A$. So, $\langle \sigma, j\rangle \models B \Rightarrow C$ and then $\langle \sigma, j\rangle \models C$. Also it is true that $\langle \sigma, j-1\rangle \models A$ and $\langle \sigma, j-1\rangle \models \bigcirc C$. This means that $\langle \sigma, j-1\rangle \models C$ because of $x : \Box((A \wedge \bigcirc C) \Rightarrow C)$. By the same argument $\langle \sigma, i\rangle \models C$ for all i such that $f(x) \leqslant i \leqslant j$. In particular $\langle \sigma, f(x)\rangle \models C$ and we are done.
(End)

Theorem 1 (Soundness of PLTL_{ND}).
Let $\mathfrak{D} = \langle \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k\rangle$ be a derivation of PLTL_{ND} *formula $\mathcal{B}$ from the set Γ. Then $\Gamma \models_{ND} B$.*

Proof. According to the definition 9, $\mathcal{A}_k$ is of the form $x : B$ for some label x. In general $x : B$ belongs to some set Λ of non-discarded PLTL_{ND} formulae of the derivation. Note that a set of non-discarded assumptions in the derivation is empty. Then by lemma 1, realisation of Γ implies realisation of Λ. In particular if the set Γ is empty, then it is realisable in arbitrary model under any mapping f^σ, by definition 5. Consequently Λ is also realisable in every model σ under any mapping f^σ. Thus, each formula in Λ is valid. In particular $x : B$ is valid. *(End)*

3.2 $\mathbf{PLTL_{ND}}$ Completeness

Theorem 2 ($\mathbf{PLTL_{ND}}$ Completeness).

Proof. We can also show that with the addition of the new rules, $\neg\Diamond$ and $\neg\mathcal{U}$, we are able to prove all the theorems of the logic PLTL. This completeness proof would be very similar to that contained in [1] being different only in establishing the fact that all the axioms of PLTL are derivable in a new system with these new rules. *(End)*

4 Discussion

We have presented a new formulation of natural deduction system for propositional linear time temporal logic and shown that the new system is sound and complete. The new formulation where few rules being replaced with only one is more elegant, and we believe this fact would be useful in possible applications and implementations. Therefore, such a topic of future research as a design of a proof-searching technique for a new system follows immediately from the results of this paper. The former system served as a background for a natural deduction system for computational temporal logic CTL[4]. Therefore, a new formulation may be treated analogously as well as it may make more efficient the proof searching procedure already designed for CTL. This is another point for future research.

References

1. A. Bolotov, A. Basukoski, O. Grigoriev, and V. Shangin. Natural deduction calculus for linear-time temporal logic. In *Joint European Conference on Artificial Intelligence (JELIA-2006)*, pages 56–68, 2006.
2. A. Bolotov, V. Bocharov, A. Gorchakov, V. Makarov, and V. Shangin. *Let Computer Prove It.* Logic and Computer. Nauka, Moscow, 2004. (In Russian), Implementation of the proof search technique for classical propositional logic available on-line at `http://prover.philos.msu.ru`.
3. A. Bolotov, V. Bocharov, A. Gorchakov, and V. Shangin. Automated first order natural deduction. In *Proceedings of IICAI*, pages 1292–1311, 2005.
4. A. Bolotov, O. Grigoriev, and V. Shangin. Natural deduction calculus for computation tree logic. In *IEEE John Vincent Atanasoff Symposium on Modern Computing*, pages 175–183, 2006.
5. M. Fisher, C. Dixon, and M. Peim. Clausal temporal resolution. *ACM Transactions on Computational Logic (TOCL)*, 1(2):12–56, 2001.
6. D. Gabbay, A. Phueli, S. Shelah, and J. Stavi. On the temporal analysis of fairness. In *Proceedings of 7th ACM Symposium on Principles of Programming Languages*, pages 163–173, Las Vegas, Nevada, 1980.
7. A. Simpson. *The Proof Theory and Semantics of Intuitionistic Modal Logic.* PhD thesis, College of Science and Engineering, School of Informatics, University of Edinburgh, 1994.