

A Syntactical Proof of the Canonical Reactivity Form for Past Linear Temporal Logic

Dimitar P. Guelev
Institute of Mathematics and Informatics,
Bulgarian Academy of Sciences
e-mail: `gelevdp@math.bas.bg`

January 4, 2008

Abstract

We present a new proof of the fact that every formula in linear temporal logic with past is equivalent to a formula of the form

$$\bigwedge_i \diamond \square \alpha_i \Rightarrow \diamond \square \beta_i,$$

where α_i and β_i are past formulas, which is known as *general canonical reactivity form*. The original proof is based on the fact that a finite automaton recognizes an *LTL*-definable ω -language iff it is counter-free, which was proved in Lenore Zuck's thesis and relies on the theorem of Krohn-Rhodes about cascade decomposition of finite automata. Unlike that, the proof presented in this paper involves only equivalence transformations of *LTL* formula and makes use of Gabbay's separation theorem, whose proof is based on equivalence transformations too. This makes it possible to obtain the canonical form without resorting to constructions outside *LTL* with past operators such as automata.

keywords: linear temporal logic, canonical reactivity normal form, separation.

Introduction

In 1989 Manna and Pnueli proposed a classification of the temporal properties of discrete-time behaviours in [MP89]. The classification applies to properties which are definable in linear temporal logic [Pnu77] with past operators (*PLTL*) and was given in three different formal settings, which were regular ω -languages and the related automata, a topological setting and the language of *PLTL*. With the class of *safety properties* being one of the base classes in the hierarchy, it is closely related to the famous *safety-liveness* classification. Most interestingly, it was shown that each of the defined classes admits a syntactical characterisation in the language of *PLTL*. The corresponding classes of *PLTL* formulas became known as *canonical safety formulas*, *canonical guarantee formulas*, etc. The hierarchy proposed in [MP89] is finite, the highest class being that of *general reactivity properties*. It was shown that every *PLTL*-definable property is a general reactivity property, and can be defined by a formula of the form

$$\bigwedge_i \diamond \square \alpha_i \Rightarrow \diamond \square \beta_i, \tag{1}$$

where α_i and β_i are past formulas. This form is known as a *general canonical reactivity form* (as opposed to just canonical reactivity, where a single implication is assumed.) Since the original paper [MP89], the hierarchy of temporal properties has drawn huge interest and inspired an impressive amount of both theoretical and applied research on the specification and verification of temporal logic requirements, not least because, along with their clear-cut mathematical properties, the proposed classes of properties have clear intuitive meaning. The monographs [MP92, MP95] have reached a very broad audience in the formal methods community.

The vast majority of the research which was inspired by [MP89, MP90] is on automata-based verification techniques, because of the straightforward correspondence between the canonical forms and the forms of the accepting conditions of finite automata for ω -languages. The original proof of the correspondence between the semantically defined classes and the respective syntactically defined classes of *PLTL* formulas involves automata too. It is based on the fact that a finite automaton recognizes a *PLTL*-definable ω -language iff it is counter-free, which was proved in Lenore Zuck's thesis [Zuc86] and relies on the theorem of Krohn-Rhodes (cf. e. g. [KR65]) about cascade decomposition of finite automata. The plan of that proof is as follows. In [Zuc86] Zuck gave a direct proof that, regardless of the type of its acceptance condition, a finite automaton accepts an ω -language which is definable by a *PLTL* formula *only if* the transition function δ of the automaton is *counter-free*, that is, if there is no set of automaton states q_0, \dots, q_{m-1} , $m \geq 2$, such that

$$q_{i+1 \bmod m} \in \delta(q_i, \sigma), \quad i = 0, \dots, m-1,$$

for some finite input word σ . For the opposite direction, it is easy to realise that an automaton is counter-free iff all of the components of its cascade decomposition according to the theorem of Krohn-Rhodes are counter-free as well. Using a cascade product satisfying this restriction, it can be shown that for a counter-free automaton a past *PLTL* formula π_q can be associated with each state q so that the regular language defined by π_q consists exactly of those input words which take the automaton from its initial state to state q . Now, given the forms of the acceptance conditions which correspond to the various classes of properties, the canonical forms can be derived immediately. For instance, in the case of safety properties the canonical formulas have the form

$$\square \left(\bigvee_{q \neq q_e} \pi_q \right),$$

where q_e is the *error* state. The general reactivity canonical form (1) was obtained using the Streett acceptance condition, which has the form $\{\langle L_1, U_1 \rangle, \dots, \langle L_n, U_n \rangle\}$, $L_i, U_i \subseteq Q$, and leads to (1) with

$$\alpha_i \rightleftharpoons \bigvee_{q \in L_i} \pi_q \quad \text{and} \quad \beta_i \rightleftharpoons \bigvee_{q \in U_i} \pi_q.$$

The possibility to prove the canonical reactivity form without automata was first demonstrated by Reynolds in [Rey00], where he outlined a syntactical proof. That proof is based on adding an auxiliary time point ∞ in order to move from $\langle \omega, < \rangle$ to a Dedekind-complete flow of time and enable the application of a separation result about *PLTL* on Dedekind-complete flows of time which can be found in [GHR94] as Theorem 10.3.20. The form (1) is obtained by applying equivalence transformations to a formula whose satisfaction at ∞ is equivalent to the satisfaction of the given formula at the beginning of time 0. The proof of Theorem 10.3.20 from [GHR94] is based on equivalence transformations too and therefore its use fits in the chosen style of establishing the canonical form (1).

In this paper we propose another syntactical proof of the canonical form (1). In our proof we work within the standard flow of time $\langle \omega, < \rangle$ and use some special cases of Gabbay's separation theorem from [Gab89], which applies to $\langle \omega, < \rangle$. A syntactical proof of this theorem can be found in [GHR94] too. The theorem states that every *PLTL* formula is equivalent to a boolean combination of past and future formulas. Among other things it enables an elegant proof of the *expressive completeness* of *PLTL*, which was first established by Kamp in [Kam68] (cf. e.g. Chapter 9 of [GHR94]). The proof of this theorem is based on equivalence transformations too and therefore its use fits in the chosen style of establishing the canonical form (1).

Structure of the paper After brief essential preliminaries we present our proof. The proof is partitioned into a sequence of lemmata which deal with increasingly special classes of *PLTL* formulas. Comments on related work are given in the concluding section.

1 Preliminaries

Here we give a brief formal introduction to propositional linear temporal logic with past (*PLTL*) and the results which we use below for the sake of self-containedness.

1.1 PLTL

The syntax of *PLTL* formulas φ can be defined by the BNF

$$\varphi ::= \perp \mid p \mid (\varphi \Rightarrow \varphi) \mid \circ\varphi \mid \bar{\circ}\varphi \mid \diamond\varphi \mid \bar{\diamond}\varphi \mid \square\varphi \mid \bar{\square}\varphi \mid (\varphi\mathbf{U}\psi) \mid (\varphi\mathbf{S}\psi)$$

where p stands for a propositional variable.

We are only interested in $\langle\omega, <\rangle$ as the flow of time for models of *PLTL* in this paper. Given a vocabulary of propositional variables \mathbf{L} , a *model for* \mathbf{L} is an infinite sequence of subsets of \mathbf{L} . The satisfaction relation \models is defined on models σ for a vocabulary \mathbf{L} , *positions* $i < \omega$ and formulas φ written in \mathbf{L} by the clauses:

$$\begin{array}{ll} s, i \not\models \perp & \\ s, i \models p & \text{iff } p \in s_i \\ s, i \models (\varphi \Rightarrow \psi) & \text{iff either } s, i \models \psi \text{ or } s, i \not\models \varphi \\ s, i \models \circ\varphi & \text{iff } s, i+1 \models \varphi \\ s, i \models \bar{\circ}\varphi & \text{iff } i > 0 \text{ and } s, i-1 \models \varphi \\ s, i \models \diamond\varphi & \text{iff } s, i+k \models \varphi \text{ for some } k < \omega \\ s, i \models \bar{\diamond}\varphi & \text{iff } s, i-k \models \varphi \text{ for some } k \leq i \\ s, i \models \square\varphi & \text{iff } s, i+k \models \varphi \text{ for all } k < \omega \\ s, i \models \bar{\square}\varphi & \text{iff } s, i-k \models \varphi \text{ for all } k \leq i \\ s, i \models (\varphi\mathbf{U}\psi) & \text{iff there is a } k < \omega \text{ such that } s, i+j \models \varphi \text{ for all } j < k \text{ and } s, i+k \models \psi \\ s, i \models (\varphi\mathbf{S}\psi) & \text{iff there is a } k \leq i \text{ such that } s, i-j \models \varphi \text{ for all } j < k \text{ and } s, i-k \models \psi \end{array}$$

The propositional connectives \neg , \vee , \wedge and \Leftrightarrow and the constant \top can be introduced as abbreviations for formulas built using \perp and \Rightarrow . Note that \diamond and \square can be defined using (\mathbf{U}) by the clauses

$$\diamond\varphi \Leftrightarrow (\top\mathbf{U}\varphi) \text{ and } \square\varphi \Leftrightarrow \neg\diamond\neg\varphi.$$

Similar clauses can be used to define $\bar{\diamond}$ and $\bar{\square}$ in terms of (\mathbf{S}) . The above clauses give the *non-strict* interpretation of (\mathbf{S}) and (\mathbf{U}) . The *strict* interpretation is as follows:

$$\begin{array}{ll} s, i \models (\varphi\mathbf{U}\psi) & \text{iff there is a } k < \omega \text{ such that } s, i+j+1 \models \varphi \text{ for all } j < k \text{ and } s, i+k+1 \models \psi \\ s, i \models (\varphi\mathbf{S}\psi) & \text{iff there is a } k < i \text{ such that } s, i-j-1 \models \varphi \text{ for all } j < k \text{ and } s, i-k-1 \models \psi \end{array}$$

Under the strict interpretation the clauses for (\mathbf{S}) and (\mathbf{U}) make no reference to the current time point. The strict interpretation enables the definition of \circ and $\bar{\circ}$ in terms of (\mathbf{U}) and (\mathbf{S}) by the clauses

$$\circ\varphi \Leftrightarrow (\perp\mathbf{U}\varphi) \text{ and } \bar{\circ}\varphi \Leftrightarrow (\perp\mathbf{S}\varphi),$$

which makes it possible to work with (\mathbf{S}) and (\mathbf{U}) as the only basic temporal operators. We also use the temporal constant \mathbf{I} , which is defined as

$$\neg(\top\mathbf{S}\top)$$

and is satisfied only at the initial time point of models based on $\langle\omega, <\rangle$.

The separation theorem that we are about to use in this paper was originally formulated for a system having only (\mathbf{S}) and (\mathbf{U}) under the strict interpretation. To be able to use it without modification, we adopt the following convention:

We use the *strict* interpretation for (\mathbf{S}) and (\mathbf{U}) and the *non-strict* interpretation for the modalities \diamond , \square , $\bar{\diamond}$ and $\bar{\square}$.

The use of different variants of the interpretation for the different modalities is not a problem. Clearly, both the strict and the non-strict variants of \diamond , \square , $\bar{\diamond}$ and $\bar{\square}$ can be defined using (\mathbf{U}) and (\mathbf{S}) with the strict interpretation.

1.2 Gabbay's separation theorem

Gabbay's theorem [Gab89] is about (\mathbf{S}) and (\mathbf{U}) as the only temporal modalities, which, under the strict interpretation, are sufficient to express all the others and, indeed, due to the expressive completeness of *PLTL*, any first-order definable modality (cf. e.g. Chapter 10 of [GHR94].) The theorem applies not just to $\langle\omega, <\rangle$, but to the linear ordering of the integers $\langle\mathbf{Z}, <\rangle$ and to finite linear flows of time as well. In the sequel *past* formulas and *future* formulas are *PLTL* formulas built without (\mathbf{U}) and without (\mathbf{S}) , respectively. The theorem is as follows:

Theorem 1 ([Gab89]) *Every PLTL formula is equivalent to a boolean combination of past and future formulas.*

Note that formulas built using only propositional connectives count as both future and past formulas in this setting. To reduce this ambiguity, the original form of the theorem states that future formulas should be of the form $\circ\varphi$. We do not need this much detail here. In our proof we also use the following more special proposition:

Lemma 1 *Let the formula φ be built using propositional variables, the formulas $(\alpha_i \mathbf{U} \beta_i)$, $i = 1, \dots, n$, propositional connectives and the past operator (\mathbf{S}) . Then φ is equivalent to a boolean combination of the formulas $(\alpha_i \mathbf{U} \beta_i)$, $i = 1, \dots, n$, and formulas built from propositional variables and the formulas α_i, β_i , $i = 1, \dots, n$, using propositional connectives and (\mathbf{S}) .*

It can be derived using the proof of Lemma 10.2.6 from Chapter 10 of [GHR94], which is part of the proof of Theorem 1. This proof is by repeated application of Lemma 10.2.5 from [GHR94] which states that formulas with only one (\mathbf{U}) -subformula $(\alpha \mathbf{U} \beta)$, in which α and β are propositional variables, have separated equivalents with $(\alpha \mathbf{U} \beta)$ being the only (\mathbf{U}) -subformula as well. Applying Lemma 10.2.5 to separate with respect to each $(\alpha_i \mathbf{U} \beta_i)$ separately by temporarily replacing the other future subformulas by propositional variables leads to a separated equivalent to φ with the above restriction. As it becomes clear below, this restriction is crucial to our proof.

2 The syntactical proof of the canonical reactivity form

Below we present our syntactical proof of the existence of an equivalent of the form (1) to every *PLTL* formula, which is the main result of this paper. Note that, unlike, e.g., separation, the equivalence holds only with respect to the beginning of time and the standard discrete time flow $\langle \omega, < \rangle$. In the sequel we use the term *conjunctive normal form* for *PLTL* formulas to denote conjunctions of disjunctions of arbitrary (\mathbf{S}) - and (\mathbf{U}) -formulas, propositional variables and their negations.

The proof starts from the observation that the equivalence

$$\varphi \Leftrightarrow \diamond \square \overline{\diamond} (\mathbf{I} \wedge \varphi)$$

holds for all *PLTL* formulas at the initial time point. Using a conjunctive normal form of a separated form of $\overline{\diamond} (\mathbf{I} \wedge \varphi)$ and the valid *PLTL* equivalence

$$\diamond \square (\psi_1 \wedge \dots \wedge \psi_n) \Leftrightarrow \diamond \square \psi_1 \wedge \dots \wedge \diamond \square \psi_n \quad (2)$$

we rewrite $\diamond \square \overline{\diamond} (\mathbf{I} \wedge \varphi)$ in the form

$$\bigwedge_{i=1}^n \diamond \square \left(\pi_i \vee \bigvee_{j=1}^{m_i} \varepsilon_{i,j} (\alpha_{i,j} \mathbf{U} \beta_{i,j}) \right)$$

where π_i are past formulas, $\alpha_{i,j}$ and $\beta_{i,j}$ are future formulas, and $\varepsilon_{i,j}$, denotes either \neg or nothing, $j = 1, \dots, m_i$, $i = 1, \dots, n$. The rest of the proof is about the transformation of formulas of the form

$$\diamond \square \left(\pi \vee \bigvee_{j=1}^m \varepsilon_j (\alpha_j \mathbf{U} \beta_j) \right) \quad (3)$$

into boolean combinations of formulas of the form $\diamond \square \pi$ where π denotes a past formula.

Lemma 2 *The following equivalences are valid in PLTL:*

$$\neg(\alpha \mathbf{U} \beta) \Leftrightarrow (\perp \mathbf{U} \square \neg \beta) \vee (\neg \beta \mathbf{U} (\neg \alpha \wedge \neg \beta)) \quad (4)$$

$$\diamond \square (\varphi \vee (\perp \mathbf{U} \square \psi)) \Leftrightarrow \diamond \square \varphi \vee \diamond \square \psi \quad (5)$$

$$\diamond \square (\psi \vee \neg(\alpha \mathbf{U} \beta)) \Leftrightarrow \diamond \square (\psi \vee (\neg \beta \mathbf{U} (\neg \alpha \wedge \neg \beta))) \vee \diamond \square \neg \beta \quad (6)$$

$$\diamond \square (\chi \vee (\varphi \mathbf{U} \psi)) \Leftrightarrow \diamond \square \chi \vee (\square \diamond \psi \wedge \diamond \square ((\neg \psi \mathbf{S} \neg \chi) \Rightarrow \varphi \vee \psi)) \quad (7)$$

Proof: The equivalences (4) and (5) are established by a direct check. Then (6) follows from (4) and (5). Here follows a proof of (7):

(\Leftarrow) If $s, n \models \diamond\Box\chi$, then obviously $s, n \models \diamond\Box(\chi \vee (\varphi\mathbf{U}\psi))$ holds too. Let $s, n \models (\Box\diamond\psi \wedge \Box((\neg\psi\mathbf{S}\neg\chi) \Rightarrow \varphi \vee \psi))$. Then $s, j \models \psi$ holds for infinitely many j and there is an i_0 such that $s, j_0 \models (\neg\psi\mathbf{S}\neg\chi) \Rightarrow \varphi \vee \psi$ for all $j_0 \geq i_0$. It is sufficient to prove that $s, j_0 \models \chi \vee (\varphi\mathbf{U}\psi)$ for $j_0 \geq i_0$ as well. Let $s, j_0 \not\models \chi$ and let j_1 be the least time point from j_0 on such that $s, j_1 + 1 \models \psi$. In case $j_1 = j_0$, we are done, because then $s, j_0 \models (\varphi\mathbf{U}\psi)$. Let $j_1 > j_0$ and $j \in (j_0, j_1]$. Then $s, j \models (\neg\psi\mathbf{S}\neg\chi)$, which implies that $s, j \models \varphi$, because $s, j \models (\neg\psi\mathbf{S}\neg\chi) \Rightarrow \varphi \vee \psi$ holds for all $j \geq i_0$. Hence $s, j \models \varphi$ for all $j \in (j_0, j_1]$. Since $s, j_1 + 1 \models \psi$, we have $s, j_0 \models (\varphi\mathbf{U}\psi)$.

(\Rightarrow) Let n be such that $s, j \models \chi \vee (\varphi\mathbf{U}\psi)$ for all $j \geq n$, and let $s, n \not\models \diamond\Box\chi$, that is, let $s, j \models \neg\chi$ for infinitely many j . Then there are infinitely many j such that $s, j \models (\varphi\mathbf{U}\psi)$, which implies that $s, n \models \Box\diamond\psi$. Let $i_0 \geq n$ and $s, i_0 \models \neg\chi$. We will establish $s, j_0 \models (\neg\psi\mathbf{S}\neg\chi) \Rightarrow \varphi \vee \psi$ for all $j_0 > i_0$. Let $s, j_0 \models (\neg\psi\mathbf{S}\neg\chi)$ and let j_1 be the greatest time point before j_0 such that $s, j_1 \models \neg\chi$. Then $s, j \models \neg\psi$ for all $j \in (j_1, j_0)$. Since $j_0 > i_0$, we have $j_1 \geq n$ and therefore $s, j_1 \models (\varphi\mathbf{U}\psi)$. This means that there exists a $k > j_1$ such that $s, k \models \psi$ and $s, j' \models \varphi$ for all $j' \in (j_1, k)$. Since $s, j \models \neg\psi$ for all $j \in (j_1, j_0)$, we have $k \geq j_0$. This implies that $s, j_0 \models \varphi \vee \psi$. \dashv

The equivalence (7) is our basic step in the elimination of (\mathbf{U}) formulas. The next lemma shows how (4), (5) and (6) help us deal with negated (\mathbf{U}) -formulas in (3).

Lemma 3 *The following equivalence is valid in PLTL:*

$$\diamond\Box \left(\pi \vee \bigvee_{j=1}^m \varepsilon_j(\alpha_j\mathbf{U}\beta_j) \right) \Leftrightarrow \diamond\Box \left(\pi \vee \bigvee_{\varepsilon_j \neq \neg} (\alpha_j\mathbf{U}\beta_j) \vee \bigvee_{\varepsilon_j = \neg} (\neg\beta_j\mathbf{U}(\neg\alpha_j \wedge \neg\beta_j)) \right) \vee \bigvee_{\varepsilon_j = \neg} \diamond\Box\neg\beta_j \quad (8)$$

Proof: Repeated application of (6). \dashv

Lemma 4 *Let π and $\alpha_i, \beta_i, i = 1, \dots, m$ be arbitrary formulas in (3). Then (3) is equivalent to a boolean combination of formulas of the form $\diamond\Box\psi$ where the ψ s are built from π and $\alpha_i, \beta_i, i = 1, \dots, m$, using only propositional connectives and (\mathbf{S}) .*

Proof: Induction on m . The disjunctive members $\diamond\Box\neg\beta_i$ from the right hand side of (8) already satisfy our requirement. Hence, after applying Lemma 3 to (3) we only need to handle the case in which all the designated occurrences of (\mathbf{U}) in (3) are positive. Let δ denote $\pi \vee \bigvee_{j=2}^m (\alpha_j\mathbf{U}\beta_j)$ for the sake of brevity. Then (3) can be written as

$$\diamond\Box(\delta \vee (\alpha_1\mathbf{U}\beta_1)). \quad (9)$$

In case $m = 1$, δ is just π and the lemma follows from (7). Otherwise, using (7) again, we establish that (9) is equivalent to

$$\diamond\Box\delta \vee (\Box\diamond\beta_1 \wedge \Box((\neg\beta_1\mathbf{S}\neg\delta) \Rightarrow \alpha_1 \vee \beta_1)) \quad (10)$$

According to the inductive hypothesis, $\diamond\Box\delta$ is equivalent to some formula ξ of the form required by the lemma. $\Box\diamond\beta_1$ is equivalent to $\neg\diamond\Box\neg\beta_1$, which has the desired form too. Now consider the (\mathbf{S}) -subformula $(\neg\beta_1\mathbf{S}\neg\delta)$ of (10). Let b and p be some fresh propositional variables and let λ stand for the substitution $[\beta_1/b, \pi/p]$. Then $(\neg\beta_1\mathbf{S}\neg\delta)$ can be written as

$$\lambda(\neg b\mathbf{S}\neg(p \vee \bigvee_{i=2}^m (\alpha_i\mathbf{U}\beta_i))).$$

According to Lemma 1, the formula next to λ above is equivalent to a boolean combination of the formulas $(\alpha_i\mathbf{U}\beta_i), i = 2, \dots, m$, and formulas built from $\alpha_i, \beta_i, i = 2, \dots, m, b$ and p using propositional connectives and (\mathbf{S}) . Let θ be the result of applying λ to such a boolean combination. Then $\diamond\Box \left(\pi \vee \bigvee_{j=1}^m (\alpha_j\mathbf{U}\beta_j) \right)$ is equivalent to

$$\xi \vee (\neg\diamond\Box\neg\beta_1 \wedge \Box(\theta \Rightarrow \alpha_1 \vee \beta_1))$$

This formula contains the subformula $\theta \Rightarrow \alpha_1 \vee \beta_1$ in the scope of $\diamond\Box$. This formula is a boolean combination of the formulas $(\alpha_i \cup \beta_i)$, $i = 2, \dots, m$, and formulas built from π and α_i and β_i , $i = 1, \dots, m$, using propositional connectives and $(.S.)$. Since $\diamond\Box$ distributes over \wedge , using conjunctive normal form, we establish that $\diamond\Box(\theta \Rightarrow \alpha_1 \vee \beta_1)$ is equivalent to a conjunction of formulas of the form

$$\diamond\Box(\eta \vee \varepsilon'_1 \gamma_1 \vee \dots \vee \varepsilon'_k \gamma_k)$$

where η is built from π and α_i and β_i , $i = 1, \dots, m$, using propositional connectives and $(.S.)$, $\gamma_1, \dots, \gamma_k \in \{(\alpha_2 \cup \beta_2), \dots, (\alpha_m \cup \beta_m)\}$, and the ε'_i s denote possible negations. Since the number of the γ_i s in each of these conjunctive members is strictly less than m , the induction hypothesis implies that these formulas have equivalents of the required form. This concludes the proof of the lemma. \dashv

The proof of the next lemma is by induction on the $(.U.)$ -depth $d_{(.U.)}$ of future formulas, which is defined by the clauses

$$\begin{aligned} d_{(.U.)}(\perp) &= d_{(.U.)}(p) = 0; \\ d_{(.U.)}(\varphi \Rightarrow \psi) &= d_{(.U.)}((\varphi S \psi)) = \max\{d_{(.U.)}(\varphi), d_{(.U.)}(\psi)\}; \\ d_{(.U.)}((\varphi U \psi)) &= \max\{d_{(.U.)}(\varphi), d_{(.U.)}(\psi)\} + 1. \end{aligned}$$

Lemma 5 *Let φ be an arbitrary separated formula. Then $\diamond\Box\varphi$ is equivalent to a boolean combination of formulas of the form $\diamond\Box\pi$ where π denotes a past formula.*

Proof: Induction on $d_{(.U.)}(\varphi)$. The lemma obviously holds in case $d_{(.U.)}(\varphi) = 0$. Using conjunctive normal form and (2), we establish that $\diamond\Box\varphi$ is equivalent to a conjunction of formulas of the form (3) where π is a past formula and the formulas $(\alpha_i \cup \beta_i)$, $i = 1, \dots, m$, occur in φ itself. Lemma 4 entails that these formulas are equivalent to boolean combinations of formulas of the form $\diamond\Box\psi$, where the ψ s are built from π and α_i, β_i , $i = 1, \dots, m$, using propositional connectives and $(.S.)$. Now Lemma 1 entails that each ψ is equivalent to a boolean combination of the $(.U.)$ -subformulas of α_i, β_i , $i = 1, \dots, m$, and formulas built from propositional variables and the arguments of these $(.U.)$ -subformulas using just propositional connectives and $(.S.)$. By repeated application of Lemma 1 to further eliminate the possible occurrences of $(.U.)$ in the scope of $(.S.)$ we establish that ψ is equivalent to a separated formula χ in which all the $(.U.)$ -subformulas are subformulas of α_i, β_i , $i = 1, \dots, m$. Since α_i, β_i , $i = 1, \dots, m$, appear in the scope of $(.U.)$ in φ , $d_{(.U.)}(\chi) < d_{(.U.)}(\varphi)$. Hence, by the induction hypothesis, $\diamond\Box\chi$ is equivalent to a boolean combination of formulas of the form $\diamond\Box\pi$ with past π . This concludes the proof of the lemma. \dashv

Using conjunctive normal form and propositional tautologies, every boolean combination of formulas of the form $\diamond\Box\pi$ with past π can be rewritten in the form

$$\bigwedge_i \left(\bigwedge_j \diamond\Box\alpha_{i,j} \Rightarrow \bigvee_k \diamond\Box\beta_{i,k} \right)$$

with $\alpha_{i,j}$ and $\beta_{i,k}$ being past formulas. That is why, having proved Lemma 5, in order to achieve the form (1), we only need to deal with the conjunctions and the disjunctions on the left and on the right of \Rightarrow , respectively. We do this by means of the valid equivalences (2) and

$$\diamond\Box\alpha \vee \diamond\Box\beta \Leftrightarrow \diamond\Box(\alpha \vee \neg(\alpha S \neg\beta)).$$

To establish the second equivalence, note that its righthand side is a *PLTL* expression for the **minex** operator from [MP89].

This completes our proof that for every *PLTL* formula φ there exists a *PLTL* formula ψ of the form (1) with the formulas α and β being past formulas.

Related work and concluding remarks

As we mentioned in the introduction, the canonical forms from [MP89] were first established using the characterization of automata which accept *PLTL*-definable ω -languages due to [Zuc86]. Another proof of the canonical form for *safety* properties and a proof of a canonical form for *liveness* properties, which is slightly different from that in [MP89], by an application of Gabbay's separation theorem was given later

in [CMP91]). A syntactical proof of the canonical reactivity form was first proposed in [Rey00]. All these proofs prescribe single applications of separation for achieving the respective canonical form. The proof from [Rey00] is based on extending the discrete flow of time by a time point ∞ and transforming the given formula to prevent the added time point from affecting its satisfaction. Our proof does not involve manipulating the flow of time, but on the other hand achieving the canonical form may require repeated applications of separation, which is known to be computationally expensive.

Discovering the various canonical forms was obviously strongly inspired by the automata connection. The syntactical proofs are examples of the possibility to technically supplant this connection by separation and this way assert the fitness of *PLTL* as a reasoning tool as opposed to being mostly a notation. Since constructing a finite state machine that recognizes the behaviours defined by a past formula is relatively straightforward, a syntactical proof of (1) implies an algorithm for the construction of ω -automata that recognize *PLTL*-defined properties as well.

Acknowledgement

The author is grateful to an anonymous referee for suggesting some important corrections to the paper.

References

- [CMP91] Edward Chang, Zohar Manna, and Amir Pnueli. The Safety-Progress Classification. In *Logic and Algebra of Specification*, NATO Advanced Science Institutes Series, pages 143–202. Springer, 1991.
- [Gab89] Dov M. Gabbay. Declarative Past and Imperative Future: Executable Temporal Logic for Interactive Systems. In *Proceedings of the Colloquium of Temporal Logic in Specification*, volume 398 of *LNCS*, pages 67–89. Springer, 1989.
- [GHR94] Dov Gabbay, Ian Hodkinson, and Mark Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects. Volume I*. Oxford University Press, 1994.
- [Kam68] J. A. W. Kamp. *Tense Logic and the Theory of Linear Order*. Ph.D. thesis, University of California, Los Angeles, 1968.
- [KR65] K. Krohn and J. Rhodes. The Algebraic Theory of Machines I. *Transactions of the American Mathematical Society*, 116:450–464, 1965.
- [MP89] Zohar Manna and Amir Pnueli. The anchored version of the temporal framework. In J.W. De Bakker, W.-P. de Roever, and G. Rozenberg, editors, *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, volume 354 of *LNCS*, pages 201–284. Springer, 1989.
- [MP90] Zohar Manna and Amir Pnueli. A Hierarchy of Temporal Properties. In *9th Symposium on Principles of Distributed Computing*, pages 377–408. ACM Press, 1990.
- [MP92] Zohar Manna and Amir Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer-Verlag, 1992.
- [MP95] Zohar Manna and Amir Pnueli. *Temporal Verification of Reactive Systems: Safety*. Springer-Verlag, 1995.
- [Pnu77] Amir Pnueli. The Temporal Logic of Programs. In *Proceedings of the 18th IEEE Symposium Foundations of Computer Science*, pages 46–57. IEEE, 1977.
- [Rey00] Mark Reynolds. More past glories. In *LICS'00: Proceedings of the 15th Annual IEEE Symposium on Logic in Computer Science*, pages 229–240, Washington, DC, USA, 2000. IEEE Computer Society.
- [Zuc86] Lenore Zuck. *Past Temporal Logic*. Ph.D. thesis, Weizmann Institute of Science, 1986.