Socratic Proofs for Propositional Linear-Time Logic. Research Report

Research report number 2(7)/2015; published: October 7, 2015.
This work is a part of the project titled *Erotetic logic in the modeling of ultimate and distributed question processing. Theoretical foundations and applications* supported by funds of the National Science Centre, Poland, (DEC-2012/04/A/HS1/00715).

Mariusz Urbański
Alexander Bolotov
Vasilyi Shangin
Oleg Grigoriev

2015
Poznań, Poland
www.kognitywistyka.amu.edu.pl/intquestpro

Socratic Proofs for Propositional Linear-Time Logic

Mariusz Urbański*
Adam Mickiewicz University, Poznań, Poland
Mariusz.Urbanski@amu.edu.pl

Vasilyi Shangin
Lomonosov Moscow State University, Russia
shangin@philos.msu.ru

Alexander Bolotov
University of Westminster, London, United Kingdom
A.Bolotov@wmin.ac.uk

Oleg Grigoriev
Lomonosov Moscow State University, Russia
grigoriev@philos.msu.ru

Abstract

In this paper we present a calculus of Socratic proofs for Propositional Linear-Time Logic.

1 Introduction

In this paper we present a calculus of Socratic proofs for Propositional Linear-Time Logic (PLTL). This logic, defined by Pnueli (1977) (see also Finger et al. (2002), Sistla and Clarke (1985)), was the first of the family of computer science-oriented logics which, in contrast to Prior-like temporal logics, are interested in computational rather than physical concept of time. The calculus we present here is based upon the semantical tableau method by Wolper (1985) and it fits into the framework of Socratic proofs by Wiśniewski (cf. Wiśniewski (2004), Wiśniewski et al. (2005) and Wiśniewski and Shangin (2006)). Similar calculi can be defined for other temporal logic – we list some of them in the last section of the report.

2 Propositional Linear-Time Logic

Propositional Linear-Time Logic (PLTL, also known as PTL, Propositional Temporal Logic), introduced in Pnueli (1977), is a propositional temporal logic

*Work on this paper was supported by funds of the National Science Centre, Poland (DEC-2012/04/A/HS1/00715).

with semantics defined on the natural numbers time (assumed model of time is discrete, linear sequence of states, finite in the past, infinite in the future).

The language of PLTL is an extension of the language of Classical Propositional Calculus (CPC). It contains at least two temporal operators: binary $\mathcal{U}$ (*until*) and unary $\bigcirc$ (*at the next moment in time*). The operators $\Box$ (*always in the future*) and $\Diamond$ (*at sometime in the future* or *eventually*) are definable if the language contains the constant $\top$ (the Truth).

The PLTL versions of $\mathcal{U}$, $\Box$, and $\Diamond$ are so-called non-strict versions of these operators. This means e.g. that a formula of the form $A\mathcal{U}B$ ('A until B') is true iff either B is true now (at, say, t_0) or B is true at some time t_i in the future and A is true at all points between t_0 and t_i, including t_0 but not t_i (in the strict version in the last clause t_0 is not included as well).

PLTL models are tuples $\mathcal{M} = \langle T, <, I \rangle$, where T is a set of states, $<$ is a binary relation over T (with usual properties) and I is an interpretation function mapping propositional variables to truth values at each state.

Semantics for temporal part of PLTL is the following:

- $\langle \mathcal{M}, t_i \rangle \models \Box A$ iff for each t_j, if $i \leq j$, then $\langle \mathcal{M}, t_j \rangle \models A$;
- $\langle \mathcal{M}, t_i \rangle \models \Diamond A$ iff there exists t_j such that $i \leq j$ and $\langle \mathcal{M}, t_j \rangle \models A$;
- $\langle \mathcal{M}, t_i \rangle \models \bigcirc A$ iff $\langle \mathcal{M}, t_{i+1} \rangle \models A$;
- $\langle \mathcal{M}, t_i \rangle \models A\mathcal{U}B$ iff there exists t_k such that $i \leq k$ and $\langle \mathcal{M}, t_k \rangle \models B$ and for all t_j such that $i \leq j < k$, $\langle \mathcal{M}, t_j \rangle \models A$.

A formula A is *PLTL-satisfiable* iff there exists $\mathcal{M}$ such that $\langle \mathcal{M}, t_0 \rangle \models A$. A formula A is *PLTL-valid* iff for each $\mathcal{M}$, $\langle \mathcal{M}, t_0 \rangle \models A$.

Different axiomatizations of PLTL exist. We present the one proposed in Gabbay et al. (1980), with the following axiom schemata:

Ax0 all CPC tautologies

Ax1 $\Box(A \to B) \to (\Box A \to \Box B)$

Ax2 $\bigcirc(A \to B) \to (\bigcirc A \to \bigcirc B)$

Ax3 $\bigcirc \neg A \leftrightarrow \neg \bigcirc A$

Ax4 $A\mathcal{U}B \leftrightarrow (B \vee (A \wedge \bigcirc A\mathcal{U}B)))$

Ax5 $\Box((B \vee (A \wedge \bigcirc C)) \to C) \to ((A\mathcal{U}B) \to C)$

Rules are: Modus Ponens (MP) and Necessitation (N) for $\Box$.

PLTL is weakly complete (it is not compact), and it has the finite model property (so it is decidable).

One of the most characteristic properties of PLTL is that most of the known non-axiomatic proof methods for this logic – and methods which are at the same time simple, effective and elegant[1] – involve looping and loop-searching. It seems that it is a rather difficult task to design a proof method without looping for logics which combines two types of temporal modalities: $\bigcirc$, which forces discrete model of time and $\mathcal{U}, \Box, \Diamond$, which can be interpreted over discrete as well as over dense models of time.

[1] That is, the methods which are based on resolution or semantic tableaux, or anything that can be interpreted in terms of finite automata.

3 Socratic proofs for PLTL – the system PLTL_T

In order to formulate PLTL_T we need to extend the language of PLTL with the following signs: $\vdash$, ?, 1 and 2. Let us call this extended language $\mathbf{P}^*$. Intuitively, $\vdash$ stands for derivability relation and ? is a question-forming operator. The numerals 1 and 2 will be used to encode tree-structure of a Socratic transformation.

There are two disjoint categories of wffs of $\mathbf{P}^*$: *declarative* wffs (d-wffs) and *erotetic* wffs (e-wffs), or questions. There are also two types of d-wffs of $\mathbf{P}^*$: *atomic* d-wffs and *indexed d-wffs*. Atomic d-wffs of $\mathbf{P}^*$ are expressions of the form $S \vdash A$, where S is a finite sequence (possibly with repetitions) of PLTL-wffs, and A is a PLTL-wff, and if A is an empty formula, then S is a non-empty sequence. Indexed d-wffs of $\mathbf{P}^*$ are expressions of the form $S \vdash^n A$ or of the form $T \vdash^n$, where $S \vdash A$ and $T \vdash$ are atomic d-wffs of $\mathbf{P}^*$ and n is a sequence of 1's or 2's, starting with 1. E-wffs, or questions of $\mathbf{P}^*$ are expressions of the form $?(\Phi)$, where Φ is a non-empty finite sequence of indexed atomic d-wffs of $\mathbf{P}^*$ (*constituents* of Φ).

In the formulation of rules we shall use the following classification of PLTL formulae to α and β types:

α	α_1	α_2	β	β_1	β_2	β_1^*
$A \wedge B$	A	B	$\neg(A \wedge B)$	$\neg A$	$\neg B$	A
$\neg(A \vee B)$	$\neg A$	$\neg B$	$A \vee B$	A	B	$\neg A$
$\neg(A \rightarrow B)$	A	$\neg B$	$A \rightarrow B$	$\neg A$	B	A
$\Box A$	A	$\bigcirc \Box A$	$\neg \Box A$	$\neg A$	$\bigcirc \Diamond \neg A$	A
$\neg \Diamond A$	$\neg A$	$\bigcirc \Box \neg A$	$\Diamond A$	A	$\bigcirc \Diamond A$	$\neg A$
$\neg(A \mathcal{U} B)$	$\neg B$	$\neg(A \wedge \bigcirc(A \mathcal{U} B))$	$A \mathcal{U} B$	B	$A \wedge \bigcirc(A \mathcal{U} B)$	$\neg B$

PT*-rules for PLTL_T:

$$\mathbf{L}_\alpha : \frac{?\,(\Phi; S \,'\, \alpha \,'\, T \vdash^n C; \Psi)}{?\,(\Phi; S \,'\, \alpha_1 \,'\, \alpha_2 \,'\, T \vdash^n C; \Psi)} \qquad \mathbf{R}_\alpha : \frac{?\,(\Phi; S \vdash^n \alpha; \Psi)}{?\,(\Phi; S \vdash^{n1} \alpha_1; S \vdash^{n2} \alpha_2; \Psi)}$$

$$\mathbf{L}_\beta : \frac{?\,(\Phi; S \,'\, \beta \,'\, T \vdash C; \Psi)}{?\,(\Phi; S \,'\, \beta_1 \,'\, T \vdash^{n1} C; S \,'\, \beta_2 \,'\, T \vdash^{n2} C; \Psi)} \qquad \mathbf{R}_\beta : \frac{?\,(\Phi; S \vdash^n \beta; \Psi)}{?\,(\Phi; S \,'\, \beta_1^* \vdash^n \beta_2; \Psi)}$$

$$\mathbf{L}_{\neg\neg} : \frac{?\,(\Phi; S \,'\, \neg\neg A \,'\, T \vdash^n C; \Psi)}{?\,(\Phi; S \,'\, A \,'\, T \vdash^n C; \Psi)} \qquad \mathbf{R}_{\neg\neg} : \frac{?\,(\Phi; S \vdash^n \neg\neg A; \Psi)}{?\,(\Phi; S \vdash^n A; \Psi)}$$

$$\mathbf{L}_{\neg\bigcirc} : \frac{?\,(\Phi; S \,'\, \neg \bigcirc A \,'\, T \vdash^n C; \Psi)}{?\,(\Phi; S \,'\, \bigcirc \neg A \,'\, T \vdash^n C; \Psi)} \qquad \mathbf{R}_{\neg\bigcirc} : \frac{?\,(\Phi; S \vdash^n \neg \bigcirc A; \Psi)}{?\,(\Phi; S \vdash^n \bigcirc \neg A; \Psi)}$$

If none of the above rules is applicable to a PLTL formula B, such a formula is called *marked*. If all PLTL-formulas within an indexed formula $S \vdash^n A$ are marked, such a formula is called *a state*.

S – P: state-prestate rule

$$\mathbf{S} - \mathbf{P} : \frac{?\,(\Phi; S \vdash^n A; \Psi)}{?\,(\Phi; S^* \vdash^n A^*; \Psi)}$$

where $S \vdash^n A$ is a state and S^* (resp. A^*) results from S (resp. A) by replacing all the formulas of the form $\bigcirc B$ with B and deleting all the remaining formulas.

Every formula of the form $S^* \vdash^m A^*$, where n is an initial subsequence of m or m is an initial subsequence of n, is called *a pre-state* (cf. Wolper (1985)).

In order to define Socratic transformation and proofs in PLTL_T we need the notions of a loop and of a loop-generating formula:

Definition 1. *Let* $\mathbf{q} = \langle Q_1, \ldots, Q_r \rangle$ *be a finite sequence of questions of* $\mathbf{P}^*$*. Let* Q_g, Q_{h-1}, Q_h $(1 \leq g < h - 1 \leq r)$ *be elements of the sequence* $\mathbf{q}$*. Let* $S_j \vdash^n A_j$ *be a constituent of* Q_g *and let* $S_k \vdash^m A_k$ *be a constituent of* Q_h *such that* $S_j = S_k$*,* $A_j = A_k$ *and the sequence* n *is an initial subsequence of the sequence* m*. Let* $S_l \vdash^i A_l$ *be a constituent of* Q_{h-1} *such that* $S_k \vdash^m A_k$ *is obtained from* $S_l \vdash^i A_l$ *by application of a PT*-rule. Then* $S_j \vdash^n A_j, \ldots, S_l \vdash^i A_l$ *form a* loop *(a sequence of atomic d-wffs of* $\mathbf{P}^*$ *... etc.), and* $S_k \vdash^m A_k$ *is called a* loop-generating formula.

Socratic transformations are sequences of questions that aim at deciding derivability of formuls from sets of formuls. Therefore, in order to define the notion of Socratic transformation, we need two conditions: starting condition (one that describes the starting point of a transformation) and how-to-proceed condition:

Definition 2. *A finite sequence* $\langle Q_1, \ldots, Q_r \rangle$ *of questions of* $\mathbf{P}^*$ *is a* Socratic transformation *of* $S \vdash A$ *iff the following conditions hold:*

(i) $Q_1 = ?(S \vdash^1 A)$*;*

(ii) Q_i *(where* $i = 2, \ldots, r$*) results from* Q_{i-1} *by applying a PT*-rule.*

Definition 3. *A constituent* ϕ *of a question* Q_i *is called* successful *iff one of the following holds:*

(a) ϕ *is of the form* $T'B'U \vdash^n B$*, or*

(b) ϕ *is of the form* $T'B'U'\neg B'W \vdash^n C$*, or*

(c) ϕ *is of the form* $T'\neg B'U'B'W \vdash^n C$*.*

Definition 4. *A Socratic transformation* $\langle Q_1, \ldots, Q_r \rangle$ *of* $S \vdash A$ *is* completed *iff the for each constituent* ϕ *of* Q_r *at least one of the following conditions hold:*

(a) no rule is applicable to PLTL-formulas in ϕ*, or*

(b) ϕ *is successful, or*

(c) ϕ *is a loop-generating formula.*

Definition 5. *A formula* B *is called an* eventuality *in* $S \vdash^n A$ *iff one of the following holds:*

(i) B *is a term of* S *and there exists a PLTL-formula* C *such that* $B = \Diamond C$*, or*

(ii) there exists a PLTL-formula C *such that* $B = A = \Box C$*.*

Definition 6. *A completed Socratic transformation* $\mathbf{q} = \langle Q_1, \ldots, Q_r \rangle$ *is a Socratic proof of* $S \vdash A$ *iff:*

(a) all the constituents of Q_n are successful, or

(b) for each non-successful constituent ϕ of Q_n, ϕ is a loop-generating formula and the loop generated by ϕ contains a pre-state with an unfulfilled eventuality.

In order to justify clause 6b observe that, because of the finite model property, a set of formulas which form a loop containing unfulfilled eventuality cannot be satisfiable. It comprises a formula saying that in some future state of the model something holds true and there is no such future state of the model. Thus from the semantical point of view the status of such loops is similar to that of successful constituents (see def. 3).

The following theorems express soundness and completeness of PLTL_T:

Theorem 1. *A formula A is PLTL-entailed by a sequence of formulae S iff there exists a Socratic proof of $S \vdash A$.*

Theorem 2. *A formula A is PLTL-valid iff there exists a Socratic proof of $\vdash A$.*

Theorem 3. *If there exists a Socratic proof of $S \vdash$, then the sequence S is inconsistent.*

Proofs of these theorems involve construction of a canonical model with maximal consistent sets of formulae as its states.

4 Examples

In the examples below by highlighting we indicate a formula which is analyzed at a current step. By double underline we indicate a formula which is a state. The question following the one containing a state is obtained by state-prestate rule.

Example 1.

1. $?(\vdash^1 \Box p \rightarrow p)$
2. $?(\Box p \vdash^1 p)$
3. $?(p, \bigcirc\Box p \vdash^1 p)$

Example 2.

1. $?(\vdash^1 \Box p \rightarrow \bigcirc p)$
2. $?(\Box p \vdash^1 \bigcirc p)$
3. $?(\underline{\underline{p, \bigcirc\Box p \vdash^1 \bigcirc p}})$
4. $?(\Box p \vdash^1 p)$
5. $?(p, \bigcirc\Box p \vdash^1 p)$

The intuitive meaning of state-prestate rule is that applying it we (semantically) move from a given state of a temporal model to the next one. The only information we are entitled to preserve then is contained in formulas of the form $\bigcirc A$: when moving from the state t_i to the state t_{i+1} we drop all the other formulas and next-time truths become present truths, so to say. Note, that because of the rules for $\Box$ operator, we do not loose information about what is always true as well.

Example 3.

1. $?(\vdash^{1} \Box(p \to \bigcirc p) \wedge p \to \Box p)$
2. $?(\Box(p \to \bigcirc p) \wedge p \vdash^{1} \Box p)$
3. $?(\Box(p \to \bigcirc p), p \vdash^{1} \Box p)\#$
4. $?(\Box(p \to \bigcirc p), p \vdash^{11} p;\ \Box(p \to \bigcirc p), p \vdash^{12} \bigcirc\Box p)$
5. $?(\Box(p \to \bigcirc p), p \vdash^{11} p;\ p \to \bigcirc p, \bigcirc\Box(p \to \bigcirc p), p \vdash^{12} \bigcirc\Box p)$
6. $?(\Box(p \to \bigcirc p), p \vdash^{11} p; \neg p, \bigcirc\Box(p \to \bigcirc p), p \vdash^{121} \bigcirc\Box p;$
   $\bigcirc p, \bigcirc\Box(p \to \bigcirc p), p \vdash^{122} \bigcirc\Box p)$
7. $?(\Box(p \to \bigcirc p), p \vdash^{11} p; \neg p, \bigcirc\Box(p \to \bigcirc p), p \vdash^{121} \bigcirc\Box p;$
   $p, \Box(p \to \bigcirc p) \vdash^{122} \Box p)\#$

Here we have an example of a loop (marked by #'s). This loop contains formulas: '$\Box(p \to \bigcirc p), p \vdash^{1} \Box p)$', '$\Box(p \to \bigcirc p), p \vdash^{12} \bigcirc\Box p)$', '$p \to \bigcirc p, \bigcirc\Box(p \to \bigcirc p), p \vdash^{12} \bigcirc\Box p)$', '$\bigcirc p, \bigcirc\Box(p \to \bigcirc p), p \vdash^{122} \bigcirc\Box p$'. The loop-generating formula is '$p, \Box(p \to \bigcirc p) \vdash^{122} \Box p$'. This is also one of the three constituents of the last question of the above transformation. Observe, that the loop generated by '$p, \Box(p \to \bigcirc p) \vdash^{122} \Box p$' contains a pre-state (namely: '$\Box(p \to \bigcirc p), p \vdash^{1} \Box p)$') with an unfulfilled eventuality (namely, '$\Box p$' right to the turnstile). As the two remaining constituents are successful, the above transformation is a Socratic proof of '$\vdash \Box(p \to \bigcirc p) \wedge p \to \Box p$'.

Example 4.

1. $?(\Box p, \Diamond\neg p \vdash^{1})\#$
2. $?(p, \bigcirc\Box p, \Diamond\neg p \vdash^{1})$
3. $?(p, \bigcirc\Box p, \neg p \vdash^{11}; p, \bigcirc\Box p, \bigcirc\Diamond\neg p \vdash^{12})$
3. $?(p, \bigcirc\Box p, \neg p \vdash^{11}; \Box p, \Diamond\neg p \vdash^{12})\#$

This is an example of consistency checking. The above transformation is a proof of '$\Box p, \Diamond\neg p \vdash$', thus the set $\{\Box p, \Diamond\neg p\}$ is inconsistent.

5 Other logics

As we mentioned at the beginning, PLTL was the first of the family of computer-science oriented temporal logics. The class of logics to which the method presented in this report is applicable comprises:

1. logics of strict versions of $\mathcal{U}$, $\Box$, and $\Diamond$;
2. propositional linear-time logics with past operators (*at the previous moment in time*, *since*, *always in the past*, *at sometime in the past*; the last three in both strict and non-strict versions);
3. logics of past and future with dense models (that is, without *next* and *previous* operators);
4. the vast class of propositional branching-time logics.

In the last case, however, a separate algorithm for loop-searching is rather indispensable in order to maintain computational effectiveness of the method.

References

Finger, M., Gabbay, D. M., and Reynolds, M. (2002). *Advanced Tense Logic*, volume 7 of *Handbook of Philosophical Logic*, pages 43–203. Springer.

Gabbay, D. M., Pnueli, A., Shelah, S., and Stavi, J. (1980). On the temporal analysis of fairness. In *7th ACM Symposium on Principles of Programming Languages*, pages 163–173.

Pnueli, A. (1977). The temporal logic of programs. In *Proceedings of the 18th Symposium on Foundations of Computer Science*, pages 46–57.

Sistla, A. P. and Clarke, E. M. (1985). The complexity of Propositional Linear Temporal Logic. *Journal of the Association for Computing Machinery*, 32(3):733–749.

Wiśniewski, A. (2004). Socratic Proofs. *Journal of Philosophical Logic*, 33(2):299–326.

Wiśniewski, A. and Shangin, V. (2006). Socratic proofs for quantifiers. *Journal of Philosophical Logic*, 35(2):147–178.

Wiśniewski, A., Vanackere, G., and Leszczyńska, D. (2005). Socratic Proofs and Paraconsistency. A Case Study. *Studia Logica*, 80:431–466.

Wolper, P. (1985). The tableau method for temporal logic: An overview. *Logique et Analyse*, 28:119–136.