Parametric Temporal Logic for "Model Measuring"

RAJEEV ALUR
University of Pennsylvania and Bell Labs
KOUSHA ETESSAMI
Bell Labs
SALVATORE LA TORRE
University of Pennsylvania and Università degli Studi di Salerno
and
DORON PELED
Bell Labs

We extend the standard model checking paradigm of linear temporal logic, LTL, to a "model measuring" paradigm where one can obtain more quantitative information beyond a "Yes/No" answer. For this purpose, we define a *parametric temporal logic*, PLTL, which allows statements such as "a request p is followed in at most x steps by a response q," where x is a free variable. We show how one can, given a formula $\varphi(x_1, \ldots, x_k)$ of PLTL and a system model K, not only determine whether there exists a valuation of $x_1, \ldots, x_k$ under which the system K satisfies the property φ, but if so find valuations which satisfy various optimality criteria. In particular, we present algorithms for finding valuations which minimize (or maximize) the maximum (or minimum) of all parameters. These algorithms exhibit the same PSPACE complexity as LTL model checking. We show that our choice of syntax for PLTL lies at the threshold of decidability for parametric temporal logics, in that several natural extensions have undecidable "model measuring" problems.

Categories and Subject Descriptors: D.2.4 [**Software Engineering**]: Software/Program Verification—*Model Checking*; F.4.1 [**Mathematical logic and Formal Languages**]: Mathematical Logic—*Temporal logic*; F.3.1 [**Logics and Meanings of Programs**]: Specifying,Verifying and Reasoning about Programs—*logics of programs*

General Terms: Theory, Verification

Additional Key Words and Phrases: Model checking, quantitative analysis, temporal logic

---

A preliminary version of this paper appeared in *ICALP'99: Proceedings of the 26th International Colloquium on Automata, Languages, and Programming*, Lecture Notes in Computer Science, vol. 1644, pp. 159-168, Springer-Verlag, 1999.
This work was partially supported by NSF CAREER award CCR97-34115, by DARPA/NASA grant NAG2-1214, by SRC contract 99-TJ-688, and by an Alfred P. Sloan Faculty Fellowship.
Authors' addresses: R. Alur and S. La Torre: Dept. of Computer and Information Science, University of Pennsylvania, 200 South 33rd St., Philadelphia, PA 19104, email: {alur,latorre}@cis.upenn.edu; K. Etessami and D. Peled: Bell Labs, 700 Mountain Ave., Murray Hill, NJ 07974, email: {kousha,doron}@research.bell-labs.com;

1. INTRODUCTION

Model checking has become a central methodology for automated verification of reactive systems (cf. [Clarke and Emerson 1981; Clarke and Kurshan 1996; Holzmann 1997]). In standard model checking, a system is described by some finite state model K and we use a temporal logic formula φ to define some required property of the behavior of K over time. We wish to check whether or not K indeed satisfies φ. The "yes/no" answer we obtain either reassures us that the system modeled by K behaves correctly, at least with respect to the property φ, or else points out a bug in our system.

Traditional temporal logics such as *linear temporal logic* (LTL) allow only *qualitative* assertions about the temporal ordering of events (for a basic understanding of temporal logic see [Pnueli 1977; Manna and Pnueli 1991; Emerson 1990]). For example, a typical temporal requirement "every request p is followed by a response q," expressed in LTL by the formula

$$\Box\ (\ p \rightarrow \Diamond\ q\ ),$$

does not specify any bound on how soon the response should follow the request. In various circumstances, for assessing the efficiency and practicality of the design being modeled, we need additional quantitative guarantees. Consequently, a variety of "real-time" or "quantitative" temporal logics have been proposed (see, for instance, [Koymans 1990; Emerson et al. 1990; Alur and Henzinger 1993; Alur et al. 1996]). A representative of such logics is *metric temporal logic* (MTL) [Koymans 1990; Alur and Henzinger 1993], which permits limiting the scope of temporal operators by subscripting them with natural numbers and a *direction* ("before" ($\leq$) or "after" ($>$)). The quantitative requirement that "every p is followed by q within 5 steps" is expressed by the MTL formula

$$\Box\ (\ p \rightarrow \Diamond_{\leq 5}\ q\ ).$$

Even with quantitative temporal logics, however, model checking still yields only a "yes/no" answer. What if we have no idea whether 5 is the best bound on the response time? We would actually like to *find out* what quantitative bounds can be placed on the eventuality. In this paper, we provide a framework for precisely this kind of "model measuring." We extend linear temporal logic to *parametric temporal logic* (PLTL), in which temporal operators can be subscripted, together with a direction, by a variable ranging over the natural numbers. In particular, $\Diamond_{\leq x}\, p$ will mean "in at most x steps p occurs" and $\Box_{\leq y}\, q$ will mean "for at least y steps q holds." Thus the variables serve to delimit the scope of the temporal operators, but make no *a priori* claim as to what their values should be.

Our response property is thus written in PLTL as

$$\Box\ (\ p \rightarrow \Diamond_{\leq x}\ q\ ).$$

The question then becomes: "for what values of x does the formula hold for the system being modeled?" For an operator such as $\Diamond_{\leq x}$, we would like to compute the minimum satisfying value of x, while for an operator such as $\Box_{\leq y}$, we would like to compute the maximum satisfying value of y. In general, a system model K and a formula $\varphi(x_1, \ldots, x_k)$ with multiple parameters define the set $V(K, \varphi)$ of valuations α of $(x_1, \ldots, x_k)$ under which φ holds for K. We present algorithms to answer various questions:

(1) Is there a parameter valuation α for which K satisfies φ? (i.e., is $V(K, \varphi)$ nonempty?)
(2) Does K satisfy φ for *any* α? (i.e., does $V(K, \varphi)$ contain all valuations?)
(3) Is the set $V(K, \varphi)$ finite (or bounded)?
(4) Find a parameter valuation $\alpha \in V(K, \varphi)$ which minimizes the maximum (or maximizes the minimum) parameter value.

The complexity of our algorithms for the above problems is essentially that of ordinary LTL model checking: PSPACE in the formula size and polynomial-time in the size of the model. When all parameterized operators in the formula are of the same polarity, we show how to compute an explicit representation of the set $V(K, \varphi)$ by symbolic constraints on parameter values. The key to our upper-bounds is a form of "pumping lemma": if a sequence satisfies a formula for large-enough values of the parameters, we can repeat (delete) appropriate cycles to obtain a sequence that satisfies the formula for larger (smaller) parameter values.

We have defined our logic with two restrictions. First, we do not allow equality subscripts (e.g., $\Diamond_{=x}$). Second, the same parameter cannot appear in association with two operators with different polarities (e.g., both $\Diamond_{\leq x}$ and $\Box_{\leq x}$). We show that our logic basically lies at the threshold of decidability for parametric temporal logics, in the sense that removing either of these two restrictions leads to an undecidable "model measuring" problem.

Related Work. The need for "parametric" temporal reasoning has been identified by various researchers. For system models with parametric bounds on delays, the reachability problem is known to be undecidable [Alur et al. 1993; Henzinger et al. 1995]. Parametric timing analysis for real-time systems based on extensions of CTL-like branching-time logics with parameters has been studied [Wang 1996; Emerson and Trefler 1999]. Algorithms for computing lower and upper bounds on delays between specified events [Courcoubetis and Yannakakis 1991], or on durations of intervals specified using LTL [Campos et al. 1996], are known and correspond to special cases of our problem. Our framework, thus, presents the first decidable formulation of multiple, possibly correlated, parametric constraints.

2. PARAMETRIC TEMPORAL LOGIC

We will first define parametric temporal logic using a full set of parameterized operators, but we will then show, in Lemma 2.2, that it suffices to consider only two basic parameterized operators in order to capture all of them. We will define our parametric temporal logic so that all formulas are already in positive normal form, where negation may only be applied to atomic propositions. Note that for any number $m \geq 0$, $\Diamond_{\leq m} \varphi$ implies $\Diamond_{\leq m+1} \varphi$, and likewise, $\Diamond_{> m+1} \varphi$ implies $\Diamond_{> m} \varphi$. In other words, these parameterized operators are monotone (either upward or downward). For reasons that will become clear, we wish to separate parameters that subscript upward-monotone operators from those that subscript downward-monotone operators.

A *parametric temporal logic* (PLTL) formula is composed, as in ordinary LTL, of atomic propositions $P = \{p_1, \ldots, p_n\}$ and their negations $\{\neg p_1, \ldots, \neg p_n\}$, the boolean connectives $\wedge$, $\vee$, the ordinary temporal operators *Next* ($\bigcirc$), *Eventually* ($\Diamond$), *Always* ($\Box$), *Until* ($\mathcal{U}$), and *Release* ($\mathcal{R}$). In addition, we define *parameterized* versions of all temporal operators except for the next, using two disjoint sets of parameter variables, $X = \{x_0, x_1, x_2, \ldots\}$ and $Y = \{y_0, y_1, y_2, \ldots\}$, along with constants $\mathbb{N} = \{0, 1, 2, \ldots\}$. For $x \in X \cup \mathbb{N}$ and $y \in Y \cup \mathbb{N}$, the parameterized operators are

$$\Diamond_{\leq x},\ \Diamond_{>y},\ \Box_{\leq y},\ \Box_{>x},\ \mathcal{U}_{\leq x}, \mathcal{U}_{>y},\ \mathcal{R}_{\leq y},\ \text{and}\ \ \mathcal{R}_{>x}\ .$$

Formulas are built up in the usual way from these operators and connectives, according to the following grammar:

$$\begin{aligned}\varphi \;:=\; & p_i|\neg p_i|\varphi \wedge \varphi|\varphi \vee \varphi|\bigcirc \varphi|\Diamond \varphi|\Diamond_{\leq x}\varphi|\Diamond_{>y}\varphi|\Box \varphi|\Box_{\leq y}\varphi|\Box_{>x}\varphi| \\ & \varphi\, \mathcal{U}\, \varphi|\varphi\, \mathcal{R}\, \varphi|\varphi \mathcal{U}_{\leq x}\varphi|\varphi \mathcal{U}_{>y}\varphi|\varphi\, \mathcal{R}_{\leq y}\, \varphi|\varphi\, \mathcal{R}_{>x}\, \varphi\end{aligned}$$

To give meaning to PLTL formulas, we interpret the variables in X and Y over the natural numbers $\mathbb{N}$. Given an assignment $\alpha : X \cup Y \rightarrow \mathbb{N}$ to the variables, along with an ω-word $w = w_0 w_1 w_2 \ldots$ over the alphabet $\Sigma = 2^P$, and a position $i \geq 0$, we inductively define what it means for (w, i, α) to satisfy a formula φ of PLTL, denoted $(w, i, \alpha) \models \varphi$. For convenience of notation, we extend α from the domain $X \cup Y$ to the domain $X \cup Y \cup \mathbb{N}$ so that, for $m \in \mathbb{N}$, $\alpha(m) = m$. The interpretation of all ordinary temporal operators and connectives is identical to their LTL interpretation. The parameterized operators are interpreted as follows:

(1) $(w, i, \alpha) \models \Diamond_{\leq x}\, \varphi$ iff $\exists j$ such that $(i \leq j \leq i + \alpha(x))$ and $(w, j, \alpha) \models \varphi$;
(2) $(w, i, \alpha) \models \Diamond_{>y}\, \varphi$ iff $\exists j$ such that $(j > i + \alpha(y))$ and $(w, j, \alpha) \models \varphi$;
(3) $(w, i, \alpha) \models \Box_{\leq y}\, \varphi$ iff $\forall j$ such that $(i \leq j \leq i + \alpha(y))$, $(w, j, \alpha) \models \varphi$;
(4) $(w, i, \alpha) \models \Box_{>x}\, \varphi$ iff $\forall j$ such that $(j > i + \alpha(x))$, $(w, j, \alpha) \models \varphi$;
(5) $(w, i, \alpha) \models \varphi\, \mathcal{U}_{\leq x}\, \psi$ iff $\exists j$ such that $(i \leq j \leq i + \alpha(x))$ and $(w, j, \alpha) \models \psi$ and $(w, i', \alpha) \models \varphi$ for all i' with $i \leq i' < j$;
(6) $(w, i, \alpha) \models \varphi\, \mathcal{U}_{>y}\, \psi$ iff $\exists j$ such that $(j > i + \alpha(y))$ and $(w, j, \alpha) \models \psi$ and $(w, i', \alpha) \models \varphi$ for all i' with $i \leq i' < j$;
(7) $(w, i, \alpha) \models \varphi\, \mathcal{R}_{\leq y}\, \psi$ iff $\forall j$ such that $(i \leq j \leq i + \alpha(y))$, either $(w, j, \alpha) \models \psi$, or $\exists i'$ such that $(i \leq i' < j)$ and $(w, i', \alpha) \models \varphi$;
(8) $(w, i, \alpha) \models \varphi\, \mathcal{R}_{>x}\, \psi$ iff $\forall j$ such that $(j > i + \alpha(x))$, either $(w, j, \alpha) \models \psi$, or $\exists i'$ such that $(i \leq i' < j)$ and $(w, i', \alpha) \models \varphi$.

We say that a parameterized operator Θ is *upward-monotone* if, for any formula φ, $\Theta_x \varphi \Rightarrow \Theta_{x+1}\varphi$ and *downward-monotone* if $\Theta_{y+1}\varphi \Rightarrow \Theta_y \varphi$. We will refer to the kind of monotonicity exhibited by a parameterized operator as its *polarity*. We now make some basic observations about these operators.

PROPOSITION 2.1. *Operators parameterized by variables in X are upward-monotone, while those parameterized by variables in Y are downward-monotone.*

*(***Duality***) For all $m \in \mathbb{N}$, the pairs $(\Diamond_{\leq m}, \Box_{\leq m})$, $(\Diamond_{>m}, \Box_{>m})$, $(\mathcal{U}_{\leq m}, \mathcal{R}_{\leq m})$, and $(\mathcal{U}_{>m}, \mathcal{R}_{>m})$ form* dual *operators, in that $(\Diamond_{\leq m}\, \varphi \Leftrightarrow \neg(\Box_{\leq m} \neg\varphi))$, $(\varphi\, \mathcal{U}_{>m}\, \psi \Leftrightarrow \neg(\neg\varphi\, \mathcal{R}_{>m}\, \neg\psi))$, and so on. Note that dual operators have opposite polarity.*

The second part of the proposition, duality, assures us that, although we do not explicitly allow negation of arbitrary formulas, our logic is indeed closed under negation. The following lemma shows, that without loss of generality, we can confine ourselves to the two parameterized operators $\Diamond_{\leq x}$ and $\Box_{\leq y}$ (while still maintaining closure under negation).

LEMMA 2.2. *The parameterized operators $\Diamond_{\leq x}$ and $\Box_{\leq y}$, along with ordinary temporal operators, suffice to define all parameterized operators and hence all of PLTL.*

PROOF. The following equivalences hold:

$$\begin{aligned}
\Diamond_{>y}\,\varphi &\equiv \Box_{\leq y}\ \Diamond\bigcirc\varphi \\
\Box_{>x}\,\varphi &\equiv \Diamond_{\leq x}\ \Box\bigcirc\varphi \\
\varphi\,\mathcal{U}_{\leq x}\,\psi &\equiv (\varphi\,\mathcal{U}\,\psi)\wedge\Diamond_{\leq x}\,\psi \\
\varphi\,\mathcal{U}_{>y}\,\psi &\equiv \Box_{\leq y}\,(\varphi\ \wedge\ \bigcirc(\varphi\,\mathcal{U}\,\psi)) \\
\varphi\,\mathcal{R}_{\leq y}\,\psi &\equiv (\varphi\,\mathcal{R}\,\psi)\vee\Box_{\leq y}\,\psi \\
\varphi\,\mathcal{R}_{>x}\,\psi &\equiv \Diamond_{\leq x}\,(\varphi\ \vee\ \bigcirc(\varphi\,\mathcal{R}\,\psi))
\end{aligned}$$

□

Henceforth, we will implicitly assume the "less than or equal" direction for the subscripts, and write $\Diamond_x$ and $\Box_y$, for $\Diamond_{\leq x}$ and $\Box_{\leq y}$, respectively. Observe, again, that the logic is closed under negation, since the operators $\Diamond_z$ and $\Box_z$ are duals of each other: $\Diamond_z\varphi \Leftrightarrow \neg\Box_z\neg\varphi$.

The careful reader might have observed that, in removing the other parameterized operators in Lemma 2.2, we might blow up the size of the formula exponentially, relative to the (parameterized) operator depth of the formula. However, this will cause *no problems* for the complexity of our subsequent algorithms, because we can view the rewritten formula not as a flat formula but as a *DAG* (where repeated subformulas correspond to the same portion of the DAG), and it is easily observed that the rewriting does not increase the number of distinct subformulas. The complexity of our algorithms will only depend on the number of distinct subformulas, and hence we have done no damage.

Let φ be a PLTL formula over P, containing parameter variables $X' \subseteq X$ and $Y' \subseteq Y$, and let $\alpha_{|_{X'\cup Y'}}$ denote the assignment α restricted to those variables. We define $S(\varphi)$ to be the set of assignments to the parameters in φ that make φ satisfiable:

$$S(\varphi) \doteq \{\alpha_{|_{X'\cup Y'}} \mid \exists w \in (2^P)^\omega \text{ such that } (w,0,\alpha) \models \varphi\}.$$

In the rest of the paper, when it is clear, we will not specify the projection and implicitly assume α defines only the variables that are relevant in the given context. Similarly, we define $V(\varphi)$ to be the set of assignments to the parameters in φ that make φ valid:

$$V(\varphi) \doteq \{\alpha \mid \forall w \in (2^P)^\omega,\ (w,0,\alpha) \models \varphi\}.$$

A *Kripke structure* $K = (Q, E, L, I)$ over the alphabet 2^P is given by a set of states Q, a transition relation $E \subseteq Q \times Q$, a labeling function $L : Q \mapsto 2^P$, and a set of initial states $I \subseteq Q$. We define a path ρ in K to be a sequence $q_0 q_1 \ldots$ of states such that $q_0 \in I$ and $E(q_i, q_{i+1})$ for all $i \geq 0$. We extend the definition of L to sequences of states, such that $L(\rho)$ is the ω-word defined by the sequence of labels on the states in ρ. Given a Kripke structure K, define $S(K, \varphi)$ to be those assignments α to the parameters in φ for which there exists a path in K that satisfies φ. In other words,

$$S(K, \varphi) \doteq \{\alpha \mid \exists \rho \in K \text{ such that } (L(\rho), 0, \alpha) \models \varphi\}.$$

Analogously, define

$$V(K, \varphi) \doteq \{\alpha \mid \forall \rho \in K,\ (L(\rho), 0, \alpha) \models \varphi\}.$$

Note that $S(\varphi)$ is the complement of $V(\neg\varphi)$, and $S(K, \varphi)$ is the complement of $V(K, \neg\varphi)$. For a set of valuations of parameters, such as $S(\varphi)$ or $V(K, \varphi)$, we are interested in several questions:

(1) **Emptiness**: Is the set empty?
(2) **Universality**: Does the set contain every valuation?
(3) **Finiteness**: Is the set finite?
(4) **Construction**: Compute an explicit representation of the set.
(5) **Optimization:** Find the *optimal* value, according to given criteria, in the set.

3. DECISION PROCEDURES

We begin by considering the fragment with only one parameterized modality, namely, $\Box_y$.

3.1 Parameterized Always

Given a set P of atomic propositions and a set Y of parameters, consider the logic PLTL$_\Box$ whose formulas are generated by the grammar

$$\varphi := p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \bigcirc \varphi \mid \Box \varphi \mid \varphi \mathcal{U} \varphi \mid \varphi \mathcal{R} \varphi \mid \Box_y \varphi \mid \Box_c \varphi \mid \Diamond_c \varphi$$

where $p \in P$, $y \in Y$, and $c \in \mathbb{N}$. A parameter valuation α for a formula φ of PLTL$_\Box$ assigns values to all the parameters in φ and $\alpha(\varphi)$ denotes the formula of PLTL$_\Box$ obtained by substituting each subscript parameter y in φ by the constant $\alpha(y)$. The formula φ defines the set $S(\varphi)$ of parameter valuations α such that the formula $\alpha(\varphi)$ is satisfiable.

3.1.1 *Closure.* For a formula φ, we use n_φ to denote the number of connectives in φ, k_φ to denote the number of parameterized connectives in φ, and c_φ to denote the product of all nonzero constants appearing in φ. Given a formula φ of the logic PLTL$_\Box$, the *closure* $cl(\varphi)$ is the set of formulas defined inductively as follows:

$cl(p) = \{p\}$; $cl(\neg p) = \{\neg p\}$;
$cl(\varphi_1 \wedge \varphi_2) = \{\varphi_1 \wedge \varphi_2\} \cup cl(\varphi_1) \cup cl(\varphi_2)$;
$cl(\varphi_1 \vee \varphi_2) = \{\varphi_1 \vee \varphi_2\} \cup cl(\varphi_1) \cup cl(\varphi_2)$;
$cl(\bigcirc \varphi) = \{\bigcirc \varphi\} \cup cl(\varphi)$;
$cl(\Box \varphi) = \{\Box \varphi\} \cup cl(\varphi)$;
$cl(\varphi_1 \mathcal{U} \varphi_2) = \{\varphi_1 \mathcal{U} \varphi_2\} \cup cl(\varphi_1) \cup cl(\varphi_2)$;
$cl(\varphi_1 \mathcal{R} \varphi_2) = \{\varphi_1 \mathcal{R} \varphi_2\} \cup cl(\varphi_1) \cup cl(\varphi_2)$;

$cl(\Box_y \varphi) = \{\Box_y \varphi\} \cup cl(\varphi)$;
$cl(\Box_c \varphi) = \{\Box_d \varphi \mid d = 1, 2, \ldots c\} \cup cl(\varphi)$;
$cl(\Diamond_c \varphi) = \{\Diamond_d \varphi \mid d = 1, 2, \ldots c\} \cup cl(\varphi)$.

Thus, the closure of a formula contains all its subformulas, and in addition, for subformulas of the form $\Box_c \varphi$ and $\Diamond_c \varphi$, we add corresponding formulas $\Box_d \varphi$ and $\Diamond_d \varphi$ for every $d < c$. The size of the closure of a formula φ is proportional to $n_\varphi c_\varphi$ and, thus, can be exponential in the description of φ. We need to consider subsets of the closure that correspond to possible truth assignments to various formulas in the closure. A subset $s \subseteq cl(\varphi)$ of the closure is said to be *consistent*[1] if whenever s contains $\Box_d \psi$ it also contains $\Box_{d'} \psi$ for $0 < d' < d$, and whenever s contains $\Diamond_d \psi$ it also contains $\Diamond_{d'} \psi$ for $d' > d$ with $\Diamond_{d'} \psi \in cl(\varphi)$. The consistency requirement reflects the monotonicity properties of the subscripted temporal operators.

LEMMA 3.1. *For a formula φ of $PLTL_\Box$, the number of consistent subsets of $cl(\varphi)$ is $O(c_\varphi 2^{n_\varphi})$.*

3.1.2 *Emptiness.* Observe that by monotonicity of the operator $\Box_y$ the sets $S(\varphi)$ and $V(\varphi)$ are downward-closed. That is, if α and β are two valuations such that $\alpha(y) \geq \beta(y)$ for each parameter $y \in Y$ and $\alpha \in S(\varphi)$, then $\beta \in S(\varphi)$ (and likewise for $V(\varphi)$). It follows that to test nonemptiness of $S(\varphi)$ and $V(\varphi)$ it suffices to check satisfiability of the formula obtained by setting each parameter to 0.

For a specific parameter valuation α, the formula $\alpha(\varphi)$ has no parameters and its satisfiability can be checked by algorithms for real-time temporal logics such as MTL. The resulting formula has temporal operators subscripted with $\leq c$, where c is a constant, and for this fragment, checking satisfiability is PSPACE and can be done in time $O(c_{\alpha(\varphi)} 2^{n_\varphi})$, by searching for fair paths in the tableau built out of consistent subsets of the closure [Alur and Henzinger 1993].

Since the logic LTL is PSPACE-hard, so is checking emptiness. This leads us to the following theorem:

THEOREM 3.2. *Given a formula φ of $PLTL_\Box$, checking emptiness of $S(\varphi)$ or $V(\varphi)$ is PSPACE-complete and can be done in time $O(c_\varphi 2^{n_\varphi})$.*

3.1.3 *Universality.* Given φ, we wish to check the universality condition on, for example, $S(\varphi)$, and therefore want to determine whether or not there is a valuation α such that the resulting formula $\alpha(\varphi)$ is unsatisfiable. For φ, we define $N_\varphi = 2 c_\varphi k_\varphi 2^{n_\varphi}$ and let α_N be the valuation that assigns N_φ to each parameter. We will establish that if α_N belongs to the set $S(\varphi)$, then so does every valuation. The next theorem is the key to our solution.

THEOREM 3.3. *Let φ be a formula of $PLTL_\Box$, α and β be parameter valuations such that for each parameter y, $\beta(y) \geq \alpha(y) \geq N_\varphi$. Then, $\alpha \in S(\varphi)$ implies $\beta \in S(\varphi)$.*

PROOF. First, we can assume that each parameter appears precisely once in φ (if a parameter y appears twice then rename the second occurrence to y' and

[1] There are many additional standard consistency requirements (e.g., if s contains $\varphi_1 \wedge \varphi_2$, it should also contain both φ_1 and φ_2). These are analogous to the tableau construction for LTL [Lichtenstein and Pnueli 1985], and while important in algorithm implementation, are not required for our counting arguments.

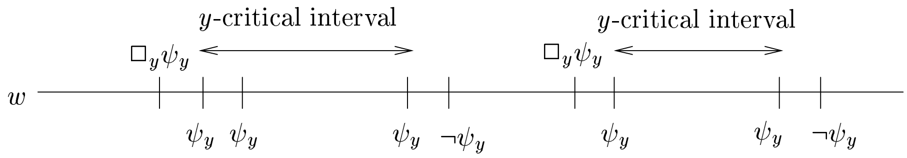

Fig. 1. Illustration of y-critical intervals

let $\alpha(y') = \alpha(y)$ and $\beta(y') = \beta(y)$). Suppose there exists a word w such that $(w, 0, \alpha) \models \varphi$. For a parameter z, define α_z such that α_z assigns the value $\alpha(z)+1$ to z and, for the remaining parameters, assigns the same value as α does. Since we can obtain β from α by a sequence of steps, where a step corresponds to incrementing one parameter by one, it suffices to prove that

$$\textit{Given } w \textit{ such that } (w, 0, \alpha) \models \varphi, \textit{ there exists } w' \textit{ such that } (w', 0, \alpha_z) \models \varphi. \quad (1)$$

For each i, define $\pi_i = \{\psi \in cl(\varphi) \mid (w, i, \alpha) \models \psi\}$ (i.e., π_i consists of those formulas in the closure that are true at position i in w according to valuation α). For each parameter y the operator $\Box_y$ occurs precisely once in φ, and let $\Box_y\psi_y$ be the corresponding subformula. An index i is said to be *y-critical* if $\Box_y\psi_y$ is in π_i, but not in π_{i+1} (see Figure 1). It follows that the formula ψ_y holds throughout the interval from position i to $i + \alpha(y)$, but not at position $i + \alpha(y) + 1$. The interval from position $i + 1$ to $i + \alpha(y)$ is said to be a *y-critical interval.* Observe the following properties of such y-critical intervals:

(1) The formula $\Box_y\psi_y$ is false at all positions in a y-critical interval.
(2) If $\Box_y\psi_y$ holds at some position j in w, then $\Box_y\psi_y$ continues to hold at subsequent positions, unless a y-critical interval is encountered.

Since by assumption for each y it holds that $\alpha(y) \geq N_\varphi$, by using the above observations along with the fact that each y-critical interval is at least $N_\varphi = 2k_\varphi c_\varphi 2^{n_\varphi}$, and there are k_φ parameters, we can obtain the following lemma.

LEMMA 3.4. *For $z \in Y$, every z-critical interval I contains a subinterval J of length $c_\varphi 2^{n_\varphi}$ such that for every $y \in Y$, either $\Box_y\psi_y$ holds at all positions in J or $\Box_y\psi_y$ holds at no position in J.*

PROOF. Let I_δ denote the initial segment of length N_φ in the z-critical interval I. We will find the desired subinterval J inside I_δ.

By the above observations, for each parameter y, the truth of $\Box_y\psi_y$ can only change at most twice in I_δ. This is so because, if $\Box_y\psi_y$ holds at some point i and does not hold at $i + 1$, then we are at the beginning of a y-critical interval, which is of length at least N_φ, and thus exceeds the length of the rest of I_δ.

Thus, there are at most $2(k_\varphi - 1)$ changes in the value of any $\Box_y\psi_y$, for any y, inside I_δ. (Remember that for $y = z$ itself, $\Box_y\psi_y$ does not change value.) Hence, I_δ is split in at most $2k_\varphi - 1$ intervals.

Now, a simple "spacing" argument establishes that there must be a subinterval of length at least $\frac{|I_\delta|}{2k_\varphi - 1}$ in which no $\Box_y\psi_y$ changes value. That establishes a mildly stronger statement than the lemma. $\Box$

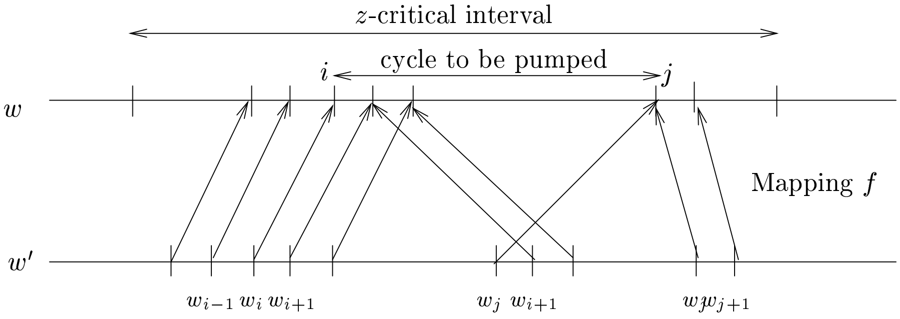

Fig. 2. Transforming w to w' by pumping cycles.

Observe that each π_i is a consistent subset of the closure, and there are at most $c_\varphi 2^{n_\varphi}$ consistent subsets. Hence, in every z-critical interval I, inside the region J defined by Lemma 3.4, we can find two positions i and j such that $\pi_i = \pi_j$ and, for every parameter y, either $\Box_y \psi_y \in \pi_l$ for all $i \le l \le j$ or $\Box_y \psi_y \notin \pi_l$ for all $i \le l \le j$. We denote this subsequence of I beginning at i and ending at j (inclusive) by $\Delta(I)$, and call it a *pumpable cycle* (there may be many pumpable cycles, but we fix $\Delta(I)$ to be, e.g., the first one to occur in the interval). Intuitively, $\Delta(I)$ is a region in which the truth of all parameterized operators is unchanged. We obtain a word w' from w by *pumping*, i.e., repeating once, $\Delta(I)$ in each z-critical interval I. For each position l in w', there is a corresponding position $f(l)$ in w. The transformation of w to w', along with the correspondence function f, is illustrated in Figure 2. Since we are pumping inside each z-critical interval and ψ_z holds at all these positions, we can use α_z, instead of α, which increments the value of the parameter z. We prove the following lemma, which will almost finish our proof of the theorem:

LEMMA 3.5. *For each $\psi \in cl(\varphi)$, for each $i \in \mathbb{N}$, if $(w, f(i), \alpha) \models \psi$ then $(w', i, \alpha_z) \models \psi$.*

PROOF. The lemma is proved by induction on the structure of the formulas in $cl(\varphi)$. The base case, when $\psi \in cl(\varphi)$ is an atomic proposition, is immediate by definition of w' and f.

Inductively, conjunctions and disjunctions are also immediate. We treat the case of each temporal operator separately.

(1) $\underline{\psi = \bigcirc\gamma}$. Assume that $(w, f(i), \alpha) \models \bigcirc\gamma$. This implies that $(w, f(i)+1, \alpha) \models \gamma$. Now, either $f(i)$ is at the right boundary of a region being pumped, or it is not. If it is not, then $f(i+1) = f(i)+1$, and, thus, by induction $(w', i+1, \alpha_z) \models \gamma$. Hence, $(w', i, \alpha_z) \models \psi$. If it is, then let k be the left boundary of that pumpable region of w. Clearly $f(i+1) = k+1$. Now, since the end points of the region being pumped agree on all subformulas, including ψ, it must be that $(w, k+1, \alpha) \models \gamma$. Thus, by induction $(w', i+1, \alpha_z) \models \gamma$, and we are done.

(2) $\underline{\psi = \Box\gamma}$. Either $f(i)$ is inside a region being pumped, or it is not. If it is not, then since $(w, f(i), \alpha) \models \Box\gamma$, for all $k \in \mathbb{N}$ it holds that $(w, f(i)+k, \alpha) \models \gamma$. Thus, by the inductive hypothesis and the definition of f, we have that $(w', i+k, \alpha_z) \models \gamma$, for all $k \in \mathbb{N}$. Hence, $(w', i, \alpha_z) \models \Box\gamma$. On the other hand, if $f(i)$ is

inside a pumped region, let $f(i)+r$ be the right boundary of this region, and let $f(i)-l$ be the left boundary. Clearly, $(w, f(i)+r, \alpha) \models \Box\,\gamma$, and since the end points of the region being pumped agree on all subformulas, $(w, f(i)-l, \alpha) \models \Box\,\gamma$. As a consequence, for every point $k \geq f(i)-l$, $(w, k, \alpha) \models \gamma$. But since, $f(j) \geq f(i)-l$ for all $j \geq i$, we have by induction that $(w', j, \alpha_z) \models \gamma$ for all $j \geq i$. Hence, $(w', i, \alpha_z) \models \Box\,\gamma$.

(3) $\underline{\psi = \delta\,\mathcal{U}\,\gamma}$. Note that this case also covers $\Diamond\,\gamma$ since $\Diamond\,\gamma \equiv (p \vee \neg p)\,\mathcal{U}\,\gamma$. Let $f(i)+k$ be the first location such that $(w, f(i)+k, \alpha) \models \gamma$, and for $0 \leq j < k$, $(w, f(i)+j, \alpha) \models \delta$. If $f(i)$ is outside a region being pumped or $f(i)+k$ is inside this region, directly by the inductive hypothesis and the definition of f we have that $(w', i+j, \alpha_z) \models \delta$ for all $0 \leq j < k$, and $(w', i+k, \alpha_z) \models \gamma$. In the other case, i.e., when $f(i)$ is inside a region being pumped and $f(i)+k$ is outside this region, we use the fact that end points of the region being pumped satisfy the same subformulas, in order to conclude that ψ must be true at both end points. Hence, by the inductive hypothesis and the definition of f, it holds that $(w', i, \alpha_z) \models \psi$.

(4) $\underline{\psi = \delta\,\mathcal{R}\,\gamma}$. This case is analogous to the previous one.

(5) $\underline{\psi = \Box_c\gamma}$. If $c = 0$, since $\Box_0\gamma = \gamma$, $(w', i, \alpha_z) \models \psi$ holds directly from the inductive hypothesis. Now, consider $c > 0$. If $f(i)$ is outside a region being pumped or $f(i)+c$ is inside this region, directly by induction and the definition of f we have that $(w', i+j, \alpha_z) \models \gamma$ for $0 \leq j \leq c$, and then, $(w', i, \alpha_z) \models \psi$. In the other case, still the inductive hypothesis, the definition of f and the fact that end points of the region being pumped satisfy the same subformulas allow us to conclude that $(w', i, \alpha_z) \models \psi$.

(6) $\underline{\psi = \Diamond_c\gamma}$. This is similar to the previous case, except that we need to make sure that the pumping will not move the first witness location where γ holds too far away from i. The only case in which this could happen is when the region being pumped is contained between $f(i)$ and $f(i)+k$, where k is the least natural number $\leq c$ such that $(w, f(i)+k, \alpha) \models \gamma$. In fact, in this case by the effect of pumping the distance between i and k could become greater than c. Fortunately, since the end points of the region being pumped satisfy the same subformulas we cannot have that a pumpable cycle stays in between $f(i)$ and $f(i)+k$.

(7) $\underline{\psi = \Box_y\psi_y}$. We now come to the last case, the parameterized operator, where we use the fact that each $\Box_y\psi_y$ holds either everywhere or nowhere in the pumpable cycles. We have to deal separately with the case where $y = z$, so first assume that $y \neq z$.
We know that $(w, f(i), \alpha) \models \psi$. Thus, for $0 \leq k \leq \alpha(y)$, $(w, f(i)+k, \alpha) \models \psi_y$ holds. Now, if there are any pumped cycles that begin between $f(i)$ and $f(i)+k$, we know inductively that the truth of ψ_y is maintained inside both copies of this cycle in w', and thus that the truth of ψ is maintained at i, and since $y \neq z$, $(w', i, \alpha_z) \models \psi$.
Now, suppose that $f(i)$ is itself inside a pumped region. From the definition of pumpable cycle, $\Box_y\psi_y$ holds in the entire cycle. Thus, ψ_y also holds in the entire cycle, and again, inductively, pumping maintains the truth of ψ_y in both copies of the cycle. Hence $(w', i, \alpha_z) \models \psi$.

Now, for the case when $y = z$. Suppose $(w, f(i), \alpha) \models \Box_z \psi_y$, and let $f(i)+r$ be the initial point of the first z-critical region after $f(i)$. Such a z-critical region must exist, unless $\Box_z \psi_y$ remains true forever after $f(i)$, in which case, for all assignments $\alpha(z)$, $(w', i, \alpha_z) \models \psi$.
Clearly ψ_y holds between $f(i)$ and $f(i)+r$ in w, and thus, inductively, also at the corresponding positions in w'. Now, we only need to note, that because we are pumping with respect to z-critical intervals, in every pumpable cycle ψ_y is true everywhere. Thus, since, inductively, pumping maintains the truth of ψ_y in both copies of the cycle, and cycles have nontrivial length, $(w', i+r, \alpha_z) \models \Box_z \psi_y$, and hence $(w', i, \alpha) \models \Box_z \psi_y$.

□

It is worth noting that pumping the cycles $\Delta(I)$ in each z-critical interval I can make more formulas true (i.e., there may be subformulas ψ and positions i such that $(w, f(i), \alpha) \not\models \psi$ but $(w', i, \alpha_z) \models \psi$).
Since $f(0) = 0$ and $(w, 0, \alpha) \models \varphi$, we conclude that $(w', 0, \alpha_z) \models \varphi$.

Since the solution set $S(\varphi)$ is downwards-closed, Theorem 3.3 implies that if $S(\varphi)$ contains the parameter valuation α_N, then it contains all the parameter valuations. Thus, checking universality reduces to checking satisfiability of the formula $\alpha_N(\varphi)$. Noting that the product of constants in $\alpha_N(\varphi)$ is $c_\varphi N_\varphi^{k_\varphi}$, from the complexity of model checking MTL, we get our desired theorem:

THEOREM 3.6. *Given a formula φ of PLTL$_\Box$, checking whether $S(\varphi)$ contains all valuations is PSPACE-complete and can be done in time $O(c_\varphi^{k_\varphi+1} k_\varphi^{k_\varphi} 2^{n_\varphi(k_\varphi+1)})$.*

3.1.4 *Finiteness.* For $i \in \{1, \ldots, k_\varphi\}$, let α_i be the valuation defined as $y_i = c_\varphi 2^{n_\varphi - k_\varphi + 1}$ and $y_j = 0$ for all $j \neq i$. Observe, that since $\Box_0 \psi \equiv \psi$, a formula $\alpha_i(\varphi)$ has $n_\varphi - k_\varphi + 1$ operators. Then, by the downward-closure of $S(\varphi)$ and Theorem 3.3, it holds that $S(\varphi)$ is infinite if and only if $\alpha_i(\varphi)$ is satisfiable for some $i \in \{1, \ldots, k_\varphi\}$. This implies the following theorem:

THEOREM 3.7. *Given a formula φ of PLTL$_\Box$, checking whether the set $S(\varphi)$ is finite is PSPACE-complete and can be done in time $O(k_\varphi c_\varphi^2 4^{n_\varphi - k_\varphi + 1})$.*

3.1.5 *Computing the Solution Set.* To compute the solution set $S(\varphi)$, first we set all parameter values to N_φ and check satisfiability of the resulting formula. Let α_{N_φ} be the parameter valuation assigning N_φ to any parameter. If the formula $\alpha_{N_\varphi}(\varphi)$ is satisfiable then $S(\varphi)$ contains all possible valuations and can be represented by the formula TRUE. If the formula $\alpha_{N_\varphi}(\varphi)$ is unsatisfiable, then we know that for a valuation α, if $\alpha(y) \geq N_\varphi$ for all parameters y, then $\alpha \notin S(\varphi)$. Thus, we need to consider only those valuations that assign a value less than N_φ to at least one parameter. We consider each parameter y and each value $c < N_\varphi$ in turn. Setting y to c gives us a formula of PLTL$_\Box$ with one parameter less, and we compute the representation of the solution set by a recursive call. The algorithm is sketched in Figure 3. The function *satisfiable* checks satisfiability of fully instantiated formulas, while the recursive function *solve* returns the representation of the set of parameter valuations satisfying φ. The representation computed by *solve* is a boolean combination of atomic formulas of the form $y = c$.

$solve(\psi)$:
$N_\psi := k_\psi \times c_\psi \times 2^{n_\psi}$;
if $satisfiable(\varphi[y_1 := N_\psi] \cdots [y_{k_\psi} := N_\psi])$ **then return** TRUE
else return $\bigvee_{i=1}^{k_\psi} \bigvee_{c=0}^{N_\psi - 1} y_i = c \wedge solve(\psi[y_i := c])$

Fig. 3. Algorithm for computing the solution set

To analyze the procedure *solve*, let $F(n_\varphi, k_\varphi, c_\varphi)$ denote the size of the representation returned by *solve*, given an input formula φ. Then, $F(n_\varphi, 0, c_\varphi) = 1$. For $k_\varphi > 0$, note that $N_\varphi = 2k_\varphi c_\varphi 2^{n_\varphi}$. In each of the $N_\varphi k_\varphi$ calls, the number of parameters is reduced by 1, and the product of constants is multiplied by at most N_φ.

$$F(n_\varphi, k_\varphi, c_\varphi) \leq c_\varphi k_\varphi^2 2^{n_\varphi} \; [\; \log(c_\varphi k_\varphi 2^{n_\varphi}) \; + \; F(n_\varphi, k_\varphi - 1, k_\varphi c_\varphi^2 2^{n_\varphi}) \;]$$

One can verify that this leads to a doubly-exponential bound: $F(n_\varphi, k_\varphi, c_\varphi) = O(k_\varphi^{2^{k_\varphi}+k_\varphi+2} c^{2^{k_\varphi}-1} 2^{n_\varphi 2^{k_\varphi+1}})$.

THEOREM 3.8. *Given a formula φ of PLTL$_\Box$, the set $S(\varphi)$ can be described by a formula which is a boolean combination of atomic formulas of the form $y = c$, for a parameter y and constant c. The size of the description is doubly exponential in number of parameter occurrences in φ.*

The structure of the solution set for a formula with $n-1$ connectives, no constants, and 2 parameters is depicted in Figure 4 ($N_\varphi = 2^n$). Note the constraints on possible solution sets. First, the solution set is downward-closed. In the case illustrated in Figure 4, $(2^n, 2^n)$ does not belong to the solution set. If it did, the solution set would include all points. Consider a horizontal line, such as $z = c < 2^n$. It either intersects with the boundary on a finite number of points, all of them with y value less than $c2^n$, or else all points on the line are contained in the solution set. A similar observation holds for vertical lines. The enumeration in the algorithm of Figure 3 does not exploit the downward-closed structure of $S(\varphi)$. It remains open whether one can effectively construct a singly-exponential representation of $S(\varphi)$. In particular, we only know an exponential lower-bound on the number of corner points in the $S(\varphi)$ region, but cannot establish a matching upper-bound, from which a singly-exponential representation would follow.

It is worth noting that, by Theorem 3.3, if the set $S(\varphi)$ is finite then it is included in the hypercube $[0, c_\varphi 2^{n_\varphi - k_\varphi + 1} - 1]^{k_\varphi}$. This leads us to the following corollary:

COROLLARY 3.9. *Let φ be a formula of PLTL$_\Box$ such that $S(\varphi)$ is finite. Then, $S(\varphi)$ can be described by a formula whose size is exponential in the number of parameters, and which is a boolean combination of atomic formulas of the form $y = c$, for a parameter y and constant c.*

3.1.6 *Optimization.* For a formula such as $\Box_y p$, we are typically interested in computing the maximum duration for which p holds and, thus, finding the maximum value of y. Given a formula with multiple parameters, we can consider various optimization functions (e.g., sum of all parameters). While all such problems can be solved by computing the explicit representation of the solution set, we present

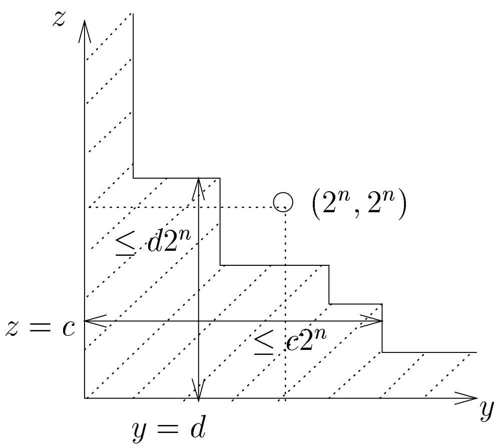

Fig. 4. Sample solution set for 2 parameters.

simpler solutions for natural problems, namely, maximizing the minimum and the maximum of all parameters.

Given φ, suppose we wish to maximize the minimum of all parameters, i.e., $\max_{\alpha \in S(\varphi)} \min_{y \in Y} \alpha(y)$. First, due to downward-closure of $S(\varphi)$, we can assume that the maximizing assignment assigns all parameters the same value. Hence, rename all parameters to a single parameter y. Then, if setting y to N_φ gives a satisfiable formula then all values of y are possible and the answer is ∞. If not, by a binary search over the interval $[0, N_\varphi]$, we can find the maximal value of c such that setting y to c gives a satisfiable formula. Thus, we need $\log N_\varphi$ satisfiability checks, each costing $O(c_\varphi^{k_\varphi+1} k_\varphi^{k_\varphi} 2^{n_\varphi(k_\varphi+1)+k_\varphi})$.

Given φ, suppose we wish to maximize the maximum of all parameters, i.e., $\max_{\alpha \in S(\varphi)} \max_{y \in Y} \alpha(y)$. Again, by downward-closure of $S(\varphi)$, we can assume that the maximizing assignment assigns the value 0 to all parameters but one. Hence, we need to compute

$$\max_y \{\alpha(y) \mid \alpha \in S(\varphi) \text{ and } \alpha(y') = 0 \text{ for } y' \neq y\}$$

So for each parameter y, we can set all others to 0 and maximize y, again by binary search.

THEOREM 3.10. *Given a formula φ of PLTL$_\Box$, maximizing the minimum of parameter values over $S(\varphi)$ can be done in time $O(|\varphi| c_\varphi^{k_\varphi+1} k_\varphi^{k_\varphi} 2^{n_\varphi(k_\varphi+1)})$, and maximizing the maximum of parameter values over $S(\varphi)$ can be done in time $O(|\varphi| c_\varphi^2 k_\varphi 2^{2n_\varphi+1})$. Both problems are PSPACE-complete.*

3.2 Parameterized Eventualities

Now we consider the logic that is the dual of PLTL$_\Box$: the logic PLTL$_\Diamond$ allows parameterized eventualities, but not parameterized always operators. Given a set P of atomic propositions and a set X of parameters, the formulas of PLTL$_\Diamond$ are

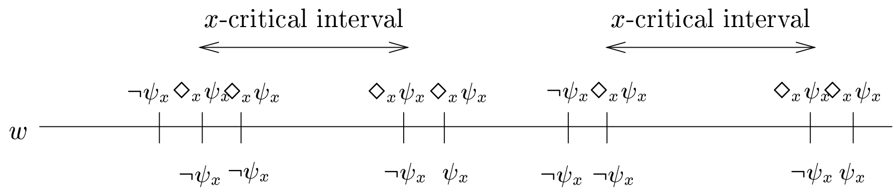

Fig. 5. Illustration of x-critical intervals.

generated by the grammar

$$\varphi := p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \bigcirc \varphi \mid \Box\varphi \mid \varphi \mathcal{U} \varphi \mid \varphi \,\mathcal{R}\, \varphi \mid \Diamond_x \varphi \mid \Box_c \varphi \mid \Diamond_c \varphi$$

where $p \in P$, $x \in X$, and $c \in \mathbb{N}$.

In this section, unless stated differently we will use the same notation as in Section 3.1. Given a formula φ of the logic PLTL$_\Diamond$, the *closure* $cl(\varphi)$ is defined as for PLTL$_\Box$, with the difference that $cl(\Box_y\varphi) = \{\Box_y\varphi\} \cup cl(\varphi)$ is substituted by $cl(\Diamond_x\varphi) = \{\Diamond_x\varphi\} \cup cl(\varphi)$.

The size of the closure of a formula φ is proportional to $n_\varphi c_\varphi$, and thus, can be exponential in the description of φ. Consistent subsets of $cl(\varphi)$ are defined as for PLTL$_\Box$ and thus, the following lemma holds.

LEMMA 3.11. *For a formula φ of PLTL$_\Diamond$, the number of consistent subsets of $cl(\varphi)$ is $O(c_\varphi 2^{n_\varphi})$.*

The results for PLTL$_\Diamond$ are analogous to those for PLTL$_\Box$. For a formula φ of PLTL$_\Diamond$, the sets $S(\varphi)$ and $V(\varphi)$ are upwards-closed. Thus, to check universality of $S(\varphi)$ or $V(\varphi)$, it suffices to check the least valuation, the one that assigns 0 to all parameters. To check $S(\varphi)$ for emptiness, we need to establish a version of Theorem 3.3 for PLTL$_\Diamond$.

THEOREM 3.12. *Let φ be a formula of PLTL$_\Diamond$, α and β be parameter valuations such that for each parameter y, $\beta(y) \geq \alpha(y) \geq N_\varphi$. Then, $\beta \in S(\varphi)$ implies $\alpha \in S(\varphi)$.*

PROOF. Since the proofs of the results we use here are very similar to those of the analogous results for PLTL$_\Box$, we omit them and just give a skeleton of the main proof.

Without loss of generality, we can assume that each parameter appears precisely once in φ. Suppose there exists a word w such that $(w, 0, \beta) \models \varphi$. For a parameter z, define β_z such that β_z assigns the value $\beta(z) - 1$ to z and, for the remaining parameters, assigns the same value as β does. Since we can obtain α from β by a sequence of steps, where a step corresponds to decrementing one parameter by one it suffices to prove that

Given a w such that $(w, 0, \beta) \models \varphi$, there exists a w' such that $(w', 0, \beta_z) \models \varphi$.

For each i, define $\pi_i = \{\psi \in cl(\varphi) \mid (w, i, \beta) \models \psi\}$ (i.e., π_i consists of those formulas in the closure that are true at position i in w according to valuation β). For each parameter x, the operator $\Diamond_x$ occurs precisely once in φ, and let $\Diamond_x\psi_x$ be the corresponding subformula. An index i is said to be *x-critical* if $\Diamond_x\psi_x$ is in π_i and ψ_x is true at $i + \beta(x)$ but not throughout the interval from position i to

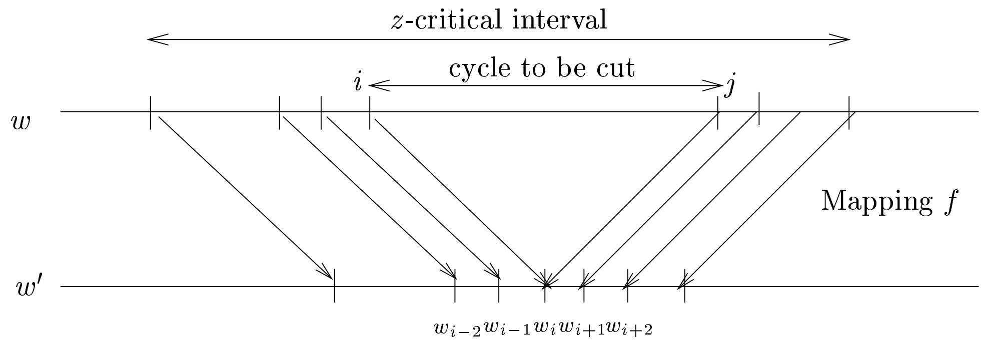

Fig. 6. Transforming w to w' by deleting cycles

$i + \beta(x) - 1$ (see Figure 5). It follows that the formula $\Diamond_x\psi_x$ holds throughout the interval from position i to $i + \beta(x)$. The interval from position i to $i + \beta(x) - 1$ is said to be a *x-critical interval* and fulfills the following properties:

(1) formula ψ_x is false at all positions in an x-critical interval and
(2) if $\Diamond_x\psi_x$ holds at some position j in w, then $\Diamond_x\psi_x$ holds at $j-1$, unless j is the left-end of an x-critical interval and ψ_x is false at $j-1$.

Since by assumption for each x it holds that $\beta(x) \geq N_\varphi$, by using the above observations along with the fact that each x-critical interval is at least $N_\varphi = 2k_\varphi c_\varphi 2^{n_\varphi}$, and there are k_φ parameters, we have the following lemma.

LEMMA 3.13. *For $z \in X$, every z-critical interval I contains a subinterval J of length $c_\varphi 2^{n_\varphi}$ such that for every parameter x, either $\Diamond_x\psi_x$ holds at all positions in J or $\Diamond_x\psi_x$ holds at no position in J.*

Observe that each such π_i is a consistent subset of the closure, and there are at most $c_\varphi 2^{n_\varphi}$ consistent subsets. Hence, in every z-critical interval I, inside the region J defined by Lemma 3.13, we can find two positions i and j such that $\pi_i = \pi_j$ and, for every parameter x, either $\Diamond_x\psi_x \in \pi_l$ for all $i \leq l \leq j$ or $\Diamond_x\psi_x \notin \pi_l$ for all $i \leq l \leq j$. We choose one of these subsequences of I and denote it by $\Gamma(I)$, and call it a *deletable cycle* (there may be many deletable cycles, but we fix $\Gamma(I)$ to be, e.g., the first one to occur in the interval). Intuitively, $\Gamma(I)$ is a region in which the truth of all parameterized operators is unchanged. We obtain a word w' from w by *deleting*, i.e., repeating once, $\Gamma(I)$ in each z-critical interval I. For each position l in w, there is a corresponding position $g(l)$ in w'. The transformation of w to w', along with the correspondence function g, is illustrated in Figure 6. Since we are deleting inside each z-critical interval and ψ_z holds at all these positions, we can use β_z, instead of β, which decrements the value of the parameter z. The following lemma holds:

LEMMA 3.14. *For any $\psi \in cl(\varphi)$ and $i \in \mathbb{N}$, if $(w, i, \beta) \models \psi$ then $(w', g(i), \beta_z) \models \psi$.*

It is worth noting that, as happened by pumping, deleting the cycles $\Gamma(I)$ in each z-critical interval I can make more formulas true (i.e., there may be subformulas ψ

and positions i such that $(w, i, \beta) \not\models \psi$ but $(w', g(i), \beta) \models \psi$). Since $g(0) = 0$ and $(w, 0, \beta) \models \varphi$, we conclude that $(w', 0, \beta_z) \models \varphi$, and we are done. $\square$

As a consequence of the theorem, to check emptiness of $S(\varphi)$, it suffices to check the valuation that assigns N_φ to all parameters. Note, that as a consequence of the upward-closed structure of $S(\varphi)$ it holds that $S(\varphi)$ is either empty or infinite. The solution set can be constructed in a similar manner, but requires atomic formulas of the form $x \geq c$. The optimization problem of interest is maximizing a parameter value, or minimizing the minimum or maximum of all parameters and can be done in PSPACE. We leave the details to the reader and summarize here:

THEOREM 3.15. *Given a formula φ of PLTL$_\Diamond$, checking emptiness and universality of the set $S(\varphi)$ is PSPACE-complete. The set $S(\varphi)$ can be described by a doubly-exponential formula which is a boolean combination of atomic formulas of the form $x \geq c$, for a parameter x and constant c. The size of the description is $O((c_\varphi k_\varphi 2^{n_\varphi})^{2^{k_\varphi}})$.*

3.3 The Full Logic PLTL

Given a formula φ of PLTL, if we instantiate values of the X-parameters (i.e., subscripts of parameterized eventualities), we get a formula of PLTL$_\Box$ and, similarly, if we instantiate values of the Y-parameters, we get a formula of PLTL$_\Diamond$. Furthermore, monotonicity properties continue to hold: if $\alpha \in S(\varphi)$, β is such that $\beta(x) \geq \alpha(x)$ for $x \in X$ and $\beta(y) \leq \alpha(y)$ for $y \in Y$, then $\beta \in S(\varphi)$. Given a valuation α for some subset Z of the parameters, the projection $[S(\varphi)]_\alpha$ of $S(\varphi)$ on α is defined to be the set of valuations $\beta \in S(\varphi)$ such that $\beta(z) = \alpha(z)$ for $z \in Z$. It follows that

(1) To check emptiness of $S(\varphi)$, we can check satisfiability of the formula resulting from setting all parameters in X to N_φ and all parameters in Y to 0. This check is PSPACE.
(2) To check universality of $S(\varphi)$, we can check satisfiability of the formula resulting from setting all parameters in X to 0 and all parameters in Y to N_φ. This check is PSPACE.
(3) For a given valuation α for the parameters in Z such that either $Z \subseteq X$ or $Z \subseteq Y$, the set $[S(\varphi)]_\alpha$ can be effectively constructed, and questions such as emptiness, universality, and minimizing/maximizing the minimum/maximum can be answered in PSPACE.

While the projections of $S(\varphi)$ when we specify all the parameters of one kind are computable, the set $S(\varphi)$ does not appear tractable, and in particular, as the results in the next section will establish, the question whether $S(\varphi)$ contains a valuation α with $\alpha(x) = \alpha(y)$ is undecidable.

3.4 Model Measuring

Model measuring corresponds to answering questions about $V(K, \varphi)$, the set of parameter valuations for which all the paths in K satisfy φ. As noted earlier, $V(K, \varphi)$ is the complement of $S(K, \neg\varphi)$.

We thus need to convert our results for $S(\varphi)$ to corresponding results about $S(K, \varphi)$. In the same way that, for LTL, model checking and satisfiability are similar problems, our answers for questions about $S(K, \varphi)$ will follow the corresponding answers for $S(\varphi)$. In particular, when φ is a $\mathrm{PLTL}_\Box$ formula, $S(K, \varphi)$ is downward-closed, so to check emptiness of $S(K, \varphi)$ we can set all parameters to 0 and check whether K satisfies the resulting formula. Similarly, when φ is a $\mathrm{PLTL}_\Diamond$ formula $S(K, \varphi)$ is upward-closed, so for checking universality we again set all parameter values to 0 and check if K satisfies the resulting formula. Likewise, Theorems 3.3 and 3.12 also hold with the modification that we have to consider paths in K and identify cycles whose end-points have identical K-states and agree on truth of all subformulas. The corresponding result is:

THEOREM 3.16. *Let φ be a formula of $PLTL_\Box$ (respectively, $PLTL_\Diamond$), K be a Kripke structure with m states, and α and β be parameter valuations such that for each parameter y, $\beta(y) \geq \alpha(y) \geq mN_\varphi$ (respectively, $\beta(y) \geq \alpha(y)$ and $\beta(y) \geq mN_\varphi$). Then, $\alpha \in S(K, \varphi)$ iff $\beta \in S(K, \varphi)$.*

From this theorem all the analogous results that we obtained for $S(\varphi)$ follow; we simply need to multiply N_φ by the number of states of the model. For instance, given K and φ of $\mathrm{PLTL}_\Box$, we can compute the maximum of the minimum parameter value over $S(K, \varphi)$ in time $O(m^{k_\varphi} c_\varphi^{k_\varphi+1} k_\varphi^{k_\varphi} 2^{n_\varphi(k_\varphi+1)})$. It is worth noting that the complexity of all the problems is polynomial in the size of the model.

4. STRONGER PARAMETERIZATION: UNDECIDABILITY

The reader might wonder why PLTL was defined as it was. In particular, why did we not allow operators of the form $\Diamond_{=x}$ and why did we not allow the same parameter variable to occur in the same formula in both an upward-monotone and downward-monotone operator? In this section we provide the answer: adding such power would make model measuring undecidable. We begin by considering what happens when you add parameterized equality operators like $\mathcal{U}_{=x}$ to the language. Let us call this language $\mathrm{PLTL}^=$.

THEOREM 4.1. *Given any two-counter Turing machine T, there are formulas φ_T and ψ_T of $PLTL^=$ such that $S(\varphi_T)$ is nonempty iff T halts and $S(\psi_T)$ is universal iff T halts.*

PROOF. We describe the formula φ_T, with parameter x. Let T have states $\{s_1, \ldots, s_k\}$, and suppose that s_k is a unique halting state. Remember that a two-counter machine can make transitions based on the current state and whether or not each of the two counters, C_1 and C_2, is 0. In making a transition to a new state, the machine can either increment or decrement a counter. A configuration of T is given by a triple (s, c_1, c_2) where s is a state, and c_1 and c_2 are respectively the current values of the first and second counters. We will use $k + 14$ propositions $\{p_1, \ldots, p_k\} \cup \{q_0^0, q_0^1\} \cup \{q_1^{-0}, q_1^0, q_1^{+0}\} \cup \{q_1^{-1}, q_1^1, q_1^{+1}\} \cup \{q_2^{-0}, q_2^0, q_2^{+0}\} \cup \{q_2^{-1}, q_2^1, q_2^{+1}\}$, in the formula φ_T. $\{p_1, \ldots, p_k\}$ will be used to denote the current state of T as the computation progresses. q_0^0 and q_0^1 will be used as separators to demarcate the boundary between consecutive configurations of T, with q_0^0 denoting the beginning of an *even* step configuration, while q_0^1 denotes the beginning of an odd step configuration, where by steps we mean steps from the initial configuration, which we consider by default to be an even step. $\{q_i^{-0}, q_i^0, q_i^{+0}\}$ and $\{q_i^{-1}, q_1^i, q_i^{+1}\}$ will be used to keep track, in unary, of the current value of counter i. The propositions

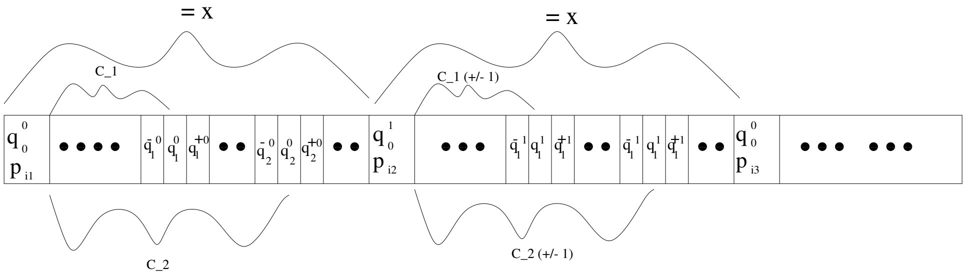

Fig. 7. ω-word encoding T's computation.

$\{q_j^{-0}, q_j^0, q_j^{+0}\}$ will be used in even numbered configurations, and $\{q_j^{-1}, q_1^j, q_j^{+1}\}$ in *odd* configurations. That is, given a T configuration (s_i, c_1, c_2), which occurs an even number of steps after the start, we can encode it (see Figure 7) by a word whose length is at least $\max(c_1, c_2) + 1$, and such that its first letter contains p_i and q_0^0, and the letters at positions c_1 and c_2 contain respectively q_1^0 and q_2^0. Thus the computation of T can be encoded by an ω-word w (see Figure 7) such that $w = w_0 w_1 \ldots$, substrings w_i are consecutive configurations of T and $|w_i|$ is larger than the maximum value stored in either counter during the entire computation. We use the parameter x to check the length of these substrings.

We will build our formula by first asserting, with ψ_1, that the configuration separators q_0^b occurs in the initial position and thereafter in precisely every x steps, alternating between q_0^0 and q_0^1, and starting with q_0^0:

$$\psi_1 \equiv q_0^0 \ \wedge \ \Box \bigwedge_b (q_0^b \to \bigcirc(\neg q_0^{\bar{b}} \, \mathcal{U}_{=x} \, q_0^{\bar{b}}) \wedge \bigcirc(\neg q_0^b \, \mathcal{U} \, q_0^{\bar{b}}))$$

Next, with ψ_2 we specify that q_1^b and q_2^b occur exactly once, for some $b \in \{0,1\}$, between consecutive occurrences of $q_0^{\{0/1\}}$'s, and that q_1^b is immediately preceded by q_1^{-b} and immediately followed by q_1^{+b}, and likewise for q_2^b. ψ_2 can easily be specified in LTL without any parameterized operators. With ψ_3 we enforce the fact that the progress of the current state, encoded by the occurrence of a unique p_i in the positions where q_0^b's occur, follows the transition relation of the Turing machine T, based on whether the zero position of the configuration contains the counter predicates q_1^b and/or q_2^b. φ_3 can also be expressed in ordinary LTL. With ψ_4, we force the fact that, depending on our current state, the counters will either be incremented by 1, remain the same, or be decremented by 1. We do this by enforcing the fact that the distance between q_1^b and the following occurrence of, respectively, $q_1^{-\bar{b}}$, $q_1^{\bar{b}}$, or $q_1^{+\bar{b}}$ is x. We leave it to the reader to write this formula using the parameterized $\mathcal{U}_{=x}$ operator. With ψ_5 we can assert what the initial configuration must look like, and in particular, since the initial configuration is an even configuration that q_1^0 and q_2^0, which maintain the counters in even configurations, must be set to 0, i.e., occur at the 0 position. Finally, with $\gamma \equiv \Diamond p_k$, we assert that the final state s_k is eventually visited. Thus, the desired formula is $\varphi_T \equiv \bigwedge_{i=1}^5 \psi_i \wedge \gamma$. For a given parameter valuation α, $(w, 0, \alpha) \models \varphi_T(x)$ if and only if w describes a halting computation of the machine T, and $\alpha(x)$ meets or exceeds the maximum value attained by either counter during the computation. For ψ_T, we use $(\bigwedge_{i=1}^4 \psi_i) \to \gamma$. $\Box$

It follows that for φ in PLTL$^=$, deciding whether $S(\varphi)$ is empty is undecidable, so is deciding whether $V(\varphi)$ is empty. Since there exists a universal Kripke structure that generates all possible ω-words over a given alphabet, the "model measuring" problems (e.g., given K and φ, is $V(K, \varphi)$ empty?) are also undecidable. The fact that introducing an "equality" subscript makes the logic hard is consistent with similar results for real-time temporal logics: for discrete-time logics, such as MTL, introduction of equality causes complexity to jump to EXPSPACE from PSPACE [Alur and Henzinger 1993], while for dense-time logics, such as MITL, introduction of equality causes complexity to jump to undecidable from decidable [Alur et al. 1996].

We now note that in the proof of Theorem 4.1 we in fact do not need to use an equality subscripts, as long as we are permitted to use the same variable x in both an upward-monotone *and* downward-monotone operator. Let PLTL$^{\uparrow\downarrow}$ denote the language in which we allow a parameter variable to occur in both such settings inside a formula. The proof of Theorem 4.1 can be adapted so that every use made there of parameterization is of the parameterized operator $\mathcal{U}_{=x}$ in the form $\neg\psi \wedge \bigcirc(\neg\psi\, \mathcal{U}_{=x}\, \psi)$, for some propositional formula ψ. Such a formula is equivalent to $\bigcirc(\neg\psi\, \mathcal{U}_{\leq x}\, \psi) \wedge (\neg\psi\, \mathcal{U}_{>x}\, \psi)$. This leads to:

COROLLARY 4.2. *Given a formula φ of PLTL$^{\uparrow\downarrow}$, checking whether $S(\varphi)$ is empty is undecidable, so is checking whether $V(\varphi)$ is empty.*

5. CONCLUSIONS

We have presented the first decidable framework for incorporating parametric reasoning in linear temporal logic, extending the "yes/no" paradigm of model checking to a "model measuring" paradigm. Given a parametric formula φ and a model K, we have shown how to find whether parameter valuations exist under which the formula φ holds for the model K, and if so to find ones which satisfy various optimality criteria. Our optimization algorithms, such as finding max-min or min-max valuations, are polynomial in the size of the model and exponential in the size of the formula. When all parameterized operators have the same monotonicity properties, we can compute an explicit representation of the solution set that is polynomial in the size of the model, exponential in the size of the formula, and doubly exponential in the number of parameters. It is an open problem whether a singly exponential representation can be computed.

ACKNOWLEDGMENTS

We thank Wolfgang Thomas, Martin Keschenau, and Jean-Francois Raskin for detailed comments on this manuscript.

REFERENCES

ALUR, R., FEDER, T., AND HENZINGER, T. 1996. The benefits of relaxing punctuality. *Journal of the ACM 43,* 1, 116–146.

ALUR, R. AND HENZINGER, T. 1993. Real-time logics: complexity and expressiveness. *Information and Computation 104,* 1, 35–77.

ALUR, R., HENZINGER, T., AND VARDI, M. 1993. Parametric real-time reasoning. In *Proceedings of the 25th ACM Symposium on Theory of Computing.* 592–601.

CAMPOS, S., CLARKE, E., AND GRUMBERG, O. 1996. Selective quantitative analysis and interval model checking. In *Proceedings of the Eighth International Conference on Computer Aided Verification*. LNCS 1102. Springer-Verlag.

CLARKE, E. AND EMERSON, E. A. 1981. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Proc. workshop on logic of programs*. LNCS, vol. 131. Springer-Verlag, 52–71.

CLARKE, E. AND KURSHAN, R. 1996. Computer-aided verification. *IEEE Spectrum 33,* 6, 61–67.

COURCOUBETIS, C. AND YANNAKAKIS, M. 1991. Minimum and maximum delay problems in real-time systems. In *Proceedings of the Third Workshop on Computer-Aided Verification*. LNCS 575. Springer-Verlag, 399–409.

EMERSON, E., MOK, A., SISTLA, A., AND SRINIVASAN, J. 1990. Quantitative temporal reasoning. In *Computer-Aided Verification, 2nd International Conference, CAV'90*. LNCS 531. Springer-Verlag, 136–145.

EMERSON, E. A. 1990. Temporal and modal logics. In *Handbook of Theoret. Comput. Sci.*, J. v. Leeuwen, Ed. Vol. B. Elsevier, Amsterdam, 995–1072.

EMERSON, E. A. AND TREFLER, R. J. 1999. Parametric quantitative temporal reasoning. In *14th IEEE Symp. on Logic in Computer Science*. 336–343.

HENZINGER, T., KOPKE, P., PURI, A., AND VARAIYA, P. 1995. What's decidable about hybrid automata. *Journal of Computer and System Sciences 25*, 94–124.

HOLZMANN, G. 1997. The model checker spin. *IEEE Trans. on Software Engineering 23,* 5, 279–295.

KOYMANS, R. 1990. Specifying real-time properties with metric temporal logic. *Journal of Real-Time Systems 2*, 255–299.

LICHTENSTEIN, O. AND PNUELI, A. 1985. Checking that finite-state concurrent programs satisfy their linear specification. In *Proceedings of the 12th ACM Symposium on Principles of Programming Languages*. 97–107.

MANNA, Z. AND PNUELI, A. 1991. *The temporal logic of reactive and concurrent systems: Specification.* Springer-verlag.

PNUELI, A. 1977. The temporal logic of programs. In *Proc. 18th Symp. on Foundations of Computer Science*. 46–57.

WANG, F. 1996. Parametric timing analysis for real-time systems. *Information and Computation 130,* 2 (1 Nov.), 131–150.

Received: December 1999; revised: October 2000; accepted: November 2000