

Model Checking the Quantitative μ -Calculus on Linear Hybrid Systems

Diana Fischer¹ and Łukasz Kaiser^{2*}

¹ Mathematische Grundlagen der Informatik, RWTH Aachen University

² CNRS & LIAFA, Université Paris Diderot – Paris 7

fischer@logic.rwth-aachen.de, kaiser@liafa.jussieu.fr

Abstract. In this work, we consider the model-checking problem for a quantitative extension of the modal μ -calculus on a class of hybrid systems. Qualitative model checking has been proved decidable and implemented for several classes of systems, but this is not the case for quantitative questions, which arise naturally in this context. Recently, quantitative formalisms that subsume classical temporal logics and additionally allow to measure interesting quantitative phenomena were introduced. We show how a powerful quantitative logic, the quantitative μ -calculus, can be model-checked with arbitrary precision on initialised linear hybrid systems. To this end, we develop new techniques for the discretisation of continuous state spaces based on a special class of strategies in model-checking games and show decidability of a class of counter-reset games that may be of independent interest.

1 Introduction

Modelling discrete-continuous systems by a hybrid of a discrete transition system and continuous variables which evolve according to a set of differential equations is widely accepted in engineering. While model-checking techniques have been applied to verify safety, liveness and other temporal properties of such systems [1, 8, 9], it is also interesting to infer quantitative values for certain queries. For example, one may not only check that a variable does not exceed a threshold, but also want to compute the maximum value of the variable over all runs.

Thus far, quantitative testing of hybrid systems has only been done by simulation, hence lacking the strong guarantees which can be given by model checking. In recent years, there has been a strong interest to extend classical model-checking techniques and logics to the quantitative setting. Several quantitative temporal logics have been introduced, see e.g. [3, 4, 6, 7, 11], together with model-checking algorithms for simple classes of systems, such as finite transition systems with discounts. Still, none of those systems allowed for dynamically changing continuous variables. We present the first model-checking algorithm for a quantitative temporal logic on a class of hybrid systems. The logic we consider, the quantitative μ -calculus [6], is based on a formalism first introduced in [4].

* Authors were supported by DFG AlgoSyn 1298 and ANR 2010 BLAN 0202 02 FREC.

It properly subsumes the standard μ -calculus, thus also CTL and LTL. Therefore the present result, namely that it is possible to model-check quantitative μ -calculus on initialised linear hybrid systems, properly generalises a previous result on model-checking LTL on such systems [8, 9], which is one of the strongest model-checking results for hybrid systems.

The logic we study allows to express properties involving suprema and infima of values of the considered variables during runs that satisfy various temporal properties, e.g. to answer “what is the maximal temperature on a run during which a safety condition holds”. To model-check formulae of the quantitative μ -calculus, we follow the classical parity game-based approach and adapt some of the methods developed in the qualitative case and for timed systems. To our surprise, these methods turned out not to be sufficient and did not easily generalise to the quantitative case. As we will show below, the quantitative systems we study behave in a substantially different way than their qualitative counterparts. We overcome this problem by first working directly with a quantitative equivalence relation, roughly similar to the region graph for timed automata, and finally introducing and solving a new kind of counter-reset games, which may be interesting in their own right.

Organisation. The organisation of this paper follows the reductions needed to model-check a formula φ over a hybrid system \mathcal{K} . In Section 2, we introduce the necessary notation, the systems and the logic. Then, we present an appropriate game model in Section 3 and show how to construct a model-checking game \mathcal{G} for the system and the formula. In Section 4, we transform the interval games constructed for arbitrary initialised linear hybrid systems to flat games, where the linear coefficients are always 1. In Section 5, we show how the strategies can be discretised and still lead to a good approximation of the original game. Finally, in Section 6, we solve the obtained parity games with counters.

$\mathcal{K}, \varphi \rightsquigarrow$ model-checking game $\mathcal{G} \rightsquigarrow$ flat $\mathcal{G} \rightsquigarrow$ counter-reset $\mathcal{G} \rightsquigarrow$ value.

2 Hybrid Systems and Quantitative Logics

We denote the real and rational numbers and integers extended with both ∞ and $-\infty$ by \mathbb{R}_∞ , \mathbb{Q}_∞ and \mathbb{Z}_∞ respectively. We write $\mathcal{I}(\mathbb{Z}_\infty)$, $\mathcal{I}(\mathbb{Q}_\infty)$ and $\mathcal{I}(\mathbb{R}_\infty)$ for all open or closed intervals over \mathbb{R}_∞ with endpoints in \mathbb{Z}_∞ , \mathbb{Q}_∞ and \mathbb{R}_∞ . For an interval $I = [i_1, i_2]$, we denote by $q \cdot I$ and $q + I$ the intervals $[q \cdot i_1, q \cdot i_2]$ and $[q + i_1, q + i_2]$, respectively, and do analogously for open intervals. We use the standard meaning of $\lfloor r \rfloor$ and $\lceil r \rceil$, and denote by $\{r\}$ the number $r - \lfloor r \rfloor$ and by $[r]$ the pair $(\lfloor r \rfloor, \lceil r \rceil)$. Hence, when writing $[r] = [s]$, we mean that r and s lie in between the same integers. Note that if $r \in \mathbb{Z}$ then $[r] = [s]$ implies that $r = s$.

Definition 1. A linear hybrid system over M variables, $\mathcal{K} = (V, E, \{P_i\}_{i \in J}, \lambda, \delta)$, is based on a directed graph (V, E) , consisting of a set of locations V and transitions $E \subseteq V \times V$. The labelling function $\lambda : E \rightarrow \mathcal{P}_{\text{fin}}(\mathcal{L}_M)$ assigns to each transition a finite set of labels. For each i of the finite index set J , the function $P_i : V \rightarrow \mathbb{R}_\infty$ assigns to each location the value of the static quantitative predicate P_i . The function $\delta : V \rightarrow \mathbb{R}^M$ assigns to each location and variable x_i the

coefficient a_i such that the variable evolves in this location according to the equation $\frac{dx_i}{dt} = a_i$. The set \mathcal{L}_M of transition labels consists of triples $l = (I, \bar{C}, R)$, where the vector \bar{C} of length M represents the constraints each of the variables need to satisfy for the transition to be allowed, the interval $I \in \mathcal{I}(\mathbb{R}_{\infty}^{\geq 0})$ represents the possible period of time that elapses before the transition is taken and the reset set R contains the indices of the variables that are reset during the transition, i.e. $i \in R$ means that x_i is set to zero.

Note that although we do not explicitly have any invariants in locations, we can simulate this by choosing either the time intervals or variable constraints on the outgoing transitions accordingly. When the values of predicates and labels range over \mathbb{Q}_{∞} or \mathbb{Z}_{∞} instead of \mathbb{R}_{∞} we talk about LHS over \mathbb{Q} and \mathbb{Z} .

The *state* of a linear hybrid system \mathcal{K} is a location combined with a valuation of all M variables, $S = V \times \mathbb{R}_{\infty}^M$. For a state $s = (v, y_1, \dots, y_M)$ we say that a transition $(v, v') \in E$ is *allowed* by a label $(I, \bar{C}, R) \in \lambda((v, v'))$ if $\bar{y} \in \bar{C}$ (i.e. if $y_i \in C_i$ for all $i = 1, \dots, M$). We say that a state $s' = (v', y'_1, \dots, y'_M)$ is a successor of s , denoted $s' \in \text{succ}(s)$, when there is a transition $(v, v') \in E$, allowed by label (I, \bar{C}, R) , such that $y'_i = 0$ for all $i \in R$ and there is a $t \in I$ such that $y'_i = y_i + (a_i \cdot t)$ where $a_i = \delta_i(v)$ for all $i \notin R \in \lambda((v, v'))$. A run of a linear hybrid system starting from location v_0 is a sequence of states s_0, s_1, \dots such that $s_0 = (v_0, 0, \dots, 0)$ and $s_{i+1} \in \text{succ}(s_i)$ for all i . Given two states s and $s' \in \text{succ}(s)$ and a reset set $R \neq \{1, \dots, M\}$ we denote by $s' -_R s$ the increase of the non-reset variables that occurred during the transition, i.e. $\frac{y'_i - y_i}{a_i}$ for some $i \notin R$ where $s = (v, \bar{y})$ and $s' = (v', \bar{y}')$.

Definition 2. A linear hybrid system \mathcal{K} is initialised if for each $(v, w) \in E$ and each variable x_i it holds that if $\delta_i(v) \neq \delta_i(w)$ then $i \in R$ for $R \in \lambda((v, w))$.

Intuitively, an initialised system cannot store the value of a variable whose evolution rate changes from one location to another.

Example 3. Consider the very simple model of a leaking gas burner depicted in Figure 1. The gas is leaking in location v_0 and not leaking in v_1 and the qualitative predicate L specifies if the leak is detected (and immediately stopped). The system has two variables, x_0 measures the time spent in the leaking location and x_1 the total elapsed time. As both variables are clocks, their coefficients are both one everywhere, i.e. the system is initialised. The time intervals indicate that a gas leak will be detected after at most one time unit and that once the gas is not leaking anymore it can only start to leak again after 30 time units.

2.1 Quantitative μ -Calculus

We use a version of the quantitative μ -calculus presented in [6] but with variables.

Definition 4. Given fixpoint variables X_j , system variables y_k and predicates $\{P_i\}_{i \in J}$, the formulae of the quantitative μ -calculus ($Q\mu$) with variables are given by the grammar:

$$\varphi ::= P_i \mid X_j \mid y_k \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \Box\varphi \mid \Diamond\varphi \mid \mu X_j.\varphi \mid \nu X_j.\varphi,$$

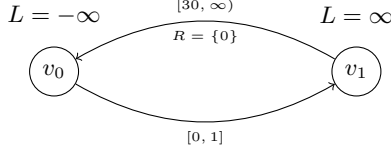


Fig. 1. Leaking gas burner

and in the cases $\mu X_j.\varphi$ and $\nu X_j.\varphi$, the variable X_j is required to appear positively in φ , i.e. under an even number of negations.

Let $\mathcal{F} = \{f : S \rightarrow \mathbb{R}_\infty\}$. Given an interpretation $\varepsilon : \mathcal{X} \rightarrow \mathcal{F}$, a variable $X \in \mathcal{X}$, and a function $f \in \mathcal{F}$, we denote by $\varepsilon[X \leftarrow f]$ the interpretation ε' , such that $\varepsilon'(X) = f$ and $\varepsilon'(X') = \varepsilon(X')$ for all $X' \neq X$.

Definition 5. Given a linear hybrid system $\mathcal{K} = (V, E, \lambda, \{P_i\}_{i \in J}, \delta)$ and an interpretation ε , a $Q\mu$ -formula yields a valuation function $\llbracket \varphi \rrbracket_\varepsilon^\mathcal{K} : S \rightarrow \mathbb{R}_\infty$ defined in the following standard way for a state $s = (v^s, y_1^s, \dots, y_M^s)$.

- $\llbracket P_i \rrbracket_\varepsilon^\mathcal{K}(s) = P_i(v^s)$, $\llbracket X \rrbracket_\varepsilon^\mathcal{K}(s) = \varepsilon(X)(s)$, and $\llbracket y_i \rrbracket_\varepsilon^\mathcal{K}(s) = y_i^s$, $\llbracket \neg \varphi \rrbracket_\varepsilon^\mathcal{K} = -\llbracket \varphi \rrbracket_\varepsilon^\mathcal{K}$
- $\llbracket \varphi_1 \wedge \varphi_2 \rrbracket_\varepsilon^\mathcal{K} = \min\{\llbracket \varphi_1 \rrbracket_\varepsilon^\mathcal{K}, \llbracket \varphi_2 \rrbracket_\varepsilon^\mathcal{K}\}$ and $\llbracket \varphi_1 \vee \varphi_2 \rrbracket_\varepsilon^\mathcal{K} = \max\{\llbracket \varphi_1 \rrbracket_\varepsilon^\mathcal{K}, \llbracket \varphi_2 \rrbracket_\varepsilon^\mathcal{K}\}$,
- $\llbracket \diamond \varphi \rrbracket_\varepsilon^\mathcal{K}(s) = \sup_{s' \in \text{succ}(s)} \llbracket \varphi \rrbracket_\varepsilon^\mathcal{K}(s')$ and $\llbracket \square \varphi \rrbracket_\varepsilon^\mathcal{K}(s) = \inf_{s' \in \text{succ}(s)} \llbracket \varphi \rrbracket_\varepsilon^\mathcal{K}(s')$,
- $\llbracket \mu X.\varphi \rrbracket_\varepsilon^\mathcal{K} = \inf\{f \in \mathcal{F} : f = \llbracket \varphi \rrbracket_{\varepsilon[X \leftarrow f]}^\mathcal{K}\}$,
- $\llbracket \nu X.\varphi \rrbracket_\varepsilon^\mathcal{K} = \sup\{f \in \mathcal{F} : f = \llbracket \varphi \rrbracket_{\varepsilon[X \leftarrow f]}^\mathcal{K}\}$.

Example 6. The formula $\mu X.(\diamond X \vee x_1)$ evaluates to the supremum of the values of x_1 on all runs from some initial state: e.g. to ∞ if evaluated on the simple leaking gas burner model. To determine the longest time period of time during which the gas is leaking undetected we use the formula $\mu X.(\diamond X \vee (x_0 \wedge L))$, which evaluates to 1 on the initial state $(v_0, \bar{0})$ in our example.

For formulae without free variables we write $\llbracket \varphi \rrbracket^\mathcal{K}$ rather than $\llbracket \varphi \rrbracket_\varepsilon^\mathcal{K}$. The remainder of this paper is dedicated to the proof of our following main result which shows that $\llbracket \varphi \rrbracket^\mathcal{K}$ can be approximated with arbitrary precision on initialised linear hybrid systems.

Theorem 7. Given an initialised linear hybrid system \mathcal{K} , a quantitative μ -calculus formula φ and an integer $n > 0$, it is decidable whether $\llbracket \varphi \rrbracket^\mathcal{K} = \infty$, $\llbracket \varphi \rrbracket^\mathcal{K} = -\infty$, and else a number $r \in \mathbb{Q}$ can be computed such that $|\llbracket \varphi \rrbracket^\mathcal{K} - r| < \frac{1}{n}$.

3 Interval Games

In this section, we define a variant of quantitative parity games suited for model checking $Q\mu$ on linear hybrid systems. This definition is a natural extension of parity games and can be viewed as a compact, finite description for a class of infinite quantitative parity games, which were introduced in [6].

Definition 8. An interval parity game (IPG) $\mathcal{G} = (V_0, V_1, E, \lambda, \delta, \iota, \Omega)$, is played on a LHS (V, E, λ, δ) and $V = V_0 \dot{\cup} V_1$ is divided into positions of either Player 0 or 1. The transition relation $E \subseteq V \times V$ describes possible moves in the game which are labelled by the function $\lambda : E \rightarrow \mathcal{P}_{\text{fin}}(\mathcal{L})$. The function $\iota : V \rightarrow M \times \mathbb{R}_{\infty} \times \mathbb{R}_{\infty}$ assigns to each position the index of a variable and a multiplicative and additive factor, which are used to calculate the payoff if a play ends in this position. The priority function $\Omega : V \rightarrow \{0, \dots, d\}$ assigns a priority to every position.

We say that the interval game is over \mathbb{Q} or \mathbb{Z} if both the underlying LHS and all constants in $\iota(v)$ are of the respective kind. Please note that this does not mean that the players have to choose their values from \mathbb{Q} or \mathbb{Z} , just that the endpoints of the intervals and constants in the payoffs are in those sets.

A state $s = (v, \bar{y}) \in V \times \mathbb{R}_{\infty}^M$ of an interval game is a position in the game graph together with a variable assignment for all M variables. A state s' is a successor of s if it is a successor in the underlying LHS, i.e. if $s' \in \text{succ}(s)$. We use the functions $\text{loc}(s) = v$ and $\text{var}(s) = \bar{y}$, $\text{var}_i(s) = y_i$ to access the components of a state. For a real number r , we denote by $r \cdot s = (v, r \cdot \text{var}_0(s), \dots, r \cdot \text{var}_M(s))$ and $r + s = (v, r + \text{var}_0(s), \dots, r + \text{var}_M(s))$. We call S_i the state set $\{s = (v, \bar{y}) : v \in V_i\}$ where player i has to move and $S = S_0 \dot{\cup} S_1$.

Intuitively, in a play of an interval parity game, the players choose successors of the current state as long as possible. The outcome $p(s_0, \dots, s_k)$ of a finite play ending in $s_k = (v, y_1, \dots, y_M)$ if $\iota(v) = (i, a, b)$ is $a \cdot y_i + b$. To improve readability, from now on we will simply write $\iota(v) = a \cdot y_i + b$ instead of $\iota(v) = (i, a, b)$. The outcome of an infinite play depends only on the lowest priority seen infinitely often in positions on the play. We will assign the value $-\infty$ to every infinite play in which the lowest priority seen infinitely often is odd, and ∞ to those, where it is even.

Formally, we use the notion from [6] and define, for an IPG with M variables $\mathcal{G} = (V_0, V_1, E, \lambda, \delta, \iota, \Omega)$, the corresponding infinite quantitative parity game without discounts $\mathcal{G}^* = (V_1 \times \mathbb{R}_{\infty}^M, V_1 \times \mathbb{R}_{\infty}^M, E^*, \lambda^*, \Omega^*)$ with $(s, s') \in E^*$ iff s' is a successor of s as above, $\Omega^*(v, \bar{z}) = \Omega(v)$ and $\lambda^*(v, \bar{z}) = \alpha \cdot z_i + \beta$ iff $\iota(v) = \alpha \cdot y_i + \beta$. The notions of plays, strategies, values and determinacy for the IPG \mathcal{G} are defined exactly as the ones for the QPG \mathcal{G}^* in [6].

3.1 Model Checking Games for $Q\mu$

A game (\mathcal{G}, v) is a model checking game for a formula φ and a system \mathcal{K}, v' , if the value of the game starting from v is exactly the value of the formula evaluated on \mathcal{K} at v' . In the qualitative case, that means, that φ holds in \mathcal{K}, v' if Player 0 wins in \mathcal{G} from v . For a linear hybrid system \mathcal{K} and a $Q\mu$ -formula φ , we construct an IPG $\text{MC}[\mathcal{K}, \varphi]$ which is the model-checking game for φ on \mathcal{K} .

The full definition of $\text{MC}[\mathcal{K}, \varphi]$ closely follows the construction presented in [6].

Intuitively, the positions are pairs consisting of a sub formula of φ and a location of \mathcal{K} . Which player moves at which position depends on the top operator

of sub formula. Player 0 moves at disjunctions to a position corresponding to one of the disjuncts and from $(\diamond\varphi, v)$ to (φ, w) where $(v, w) \in E^{\mathcal{K}}$, and Player 1 makes analogous moves for conjunctions and \square . From fixed-point variables the play moves back to the defining formula and the priorities of positions depends on the alternation level of fixed points, assigning odd priorities to least fixed points and even priorities to greatest fixed points.

Example 9. We continue our example of the leaking gas burner and present in Figure 2 the model checking game for the previously introduced system and formula. In this interval parity game, ellipses depict positions of Player 0 and rectangles those of Player 1. In this game, all priorities are odd (and therefore omitted), i.e. infinite plays are bad for Player 0. As in the underlying system, there are no constraints on the variables and only in two moves a time unit can be picked by Player 0. In terminal nodes, either the variable x_0 or the predicate L is evaluated for the payoff. The value of the game is 1 (as is the value of the formula on the system starting from either node) and an optimal strategy for Player 0 is picking 1 from $[0, 1]$ and then leaving the cycle where Player 1 is forced to choose between the evaluation of x_0 or L at v_1 . Since he is minimising, he will choose to evaluate x_0 .

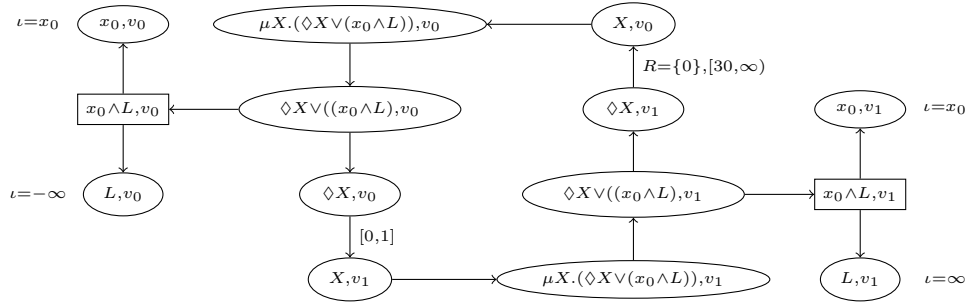


Fig. 2. Model checking game for $\mu X.(\diamond X \vee (x_0 \wedge L))$ on leaking gas burner.

It has been shown in [6] that quantitative parity games of any size are determined and that they are model checking games for $Q\mu$. These results translate to interval parity games and we can conclude the following.

Theorem 10. *Every interval parity game is determined and for every formula φ in $Q\mu$, linear hybrid system \mathcal{K} , and a location v of \mathcal{K} , it holds that*

$$\text{valMC}[\mathcal{K}, \varphi]((\varphi, v), \bar{0}) = \llbracket \varphi \rrbracket^{\mathcal{K}}(v, \bar{0}).$$

4 Basic Properties of Interval Games

At first sight, interval games seem to be very similar to timed games. Simple timed games are solved by playing on the region graph and can thus be discrete-

tised. To stress that quantitative payoffs indeed make a difference, we present in Figure 3 an initialised interval parity game with the interesting property that it is not enough to play integer values, even though the underlying system is over \mathbb{Z}_∞ . This simple game contains only one variable (a clock) and has no constraints on the variables in any of the transitions so only the time intervals are shown. Also, as infinite plays are not possible, the priorities are omitted, as well as the indices of non-terminal positions. This game illustrates that it may not be optimal to play integer values since choosing time $\frac{1}{2}$ in the first move is optimal for Player 0. This move guarantees an outcome of $-\frac{1}{2}$ which is equal to the value of the game.

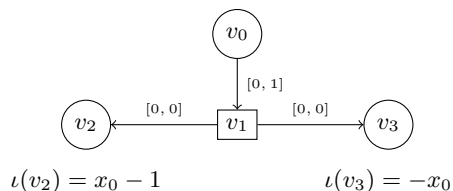


Fig. 3. Game with integer coefficients and non-integer value.

4.1 Flattening Initialised Interval Games

So far, we have considered games where the values of variables can change at different rates during the time spent in locations. In this section, we show that for initialised games it is sufficient to look at easier games where all rates are one, similar to timed games but with more complex payoff rules. We call these games flat and show that for every initialised IPG we can construct a flat IPG with the same value. To do so, we have to consider the regions where the coefficients do not change and rescale the constraints and payoffs accordingly.

Definition 11. *An interval parity game $\mathcal{G} = (V_0, V_1, E, \lambda, \delta, \iota, \Omega)$ is flat if and only if $\delta_i(v, \bar{x}) = 1$ for all $v \in V$ and $i = 1 \dots M$.*

Lemma 12. *For each initialised interval parity game \mathcal{G} there exists a flat game \mathcal{G}' with the same value.*

Consequently, from now on we only consider flat interval parity games and therefore omit the coefficients, as they are all equal to one.

4.2 Multiplying Interval Games

Definition 13. *For a flat IPG $\mathcal{G} = (V_0, V_1, E, \lambda, \iota, \Omega)$ and a value $q \in \mathbb{Q}$, we denote by $q \cdot \mathcal{G} = (V, E, \lambda', \iota', \Omega)$ the IPG where $\iota'(v) = a \cdot x_i + q \cdot b$ iff $\iota(v) = a \cdot x_i + b$ for all $v \in V$, and $(I', \bar{C}', R) \in \lambda'((v, w))$ iff $(I, \bar{C}, R) \in \lambda((v, w))$ with $I' = q \cdot I$ and $C'_i = q \cdot C_i$ for all $(v, w) \in E$.*

Intuitively, this means that all endpoints in the time intervals (open and closed), in the constraints, and all additive values in the payoff function ι are multiplied by q . The values of $q \cdot \mathcal{G}$ are also equal to the values of \mathcal{G} multiplied by q .

Lemma 14. *For every IPG \mathcal{G} over \mathbb{Q}_∞ and $q \in \mathbb{Q}, q \neq 0$ it holds in all states s that $q \cdot \text{val}\mathcal{G}(s) = \text{val } q \cdot \mathcal{G}(q \cdot s)$.*

Note that all multiplicative factors in ι are the same in \mathcal{G} and in $q \cdot \mathcal{G}$. Moreover, if we multiply all constants in ι in a game \mathcal{G} (both the multiplicative and the additive ones) by a positive value r , then the value of \mathcal{G} will be multiplied by r , by an analogous argument as above. Thus, if we first take r as the least common multiple of all denominators of multiplicative factors in ι and multiply all ι constants as above, and then take q as the least common multiple of all denominators of endpoints in the intervals and additive factors in the resulting game \mathcal{G} and build $q \cdot \mathcal{G}$, we can conclude the following.

Corollary 15. *For every finite IPG \mathcal{G} over \mathbb{Q}_∞ , there exists an IPG \mathcal{G}' over \mathbb{Z}_∞ and $q, r \in \mathbb{Z}$ such that $\text{val}\mathcal{G}(s) = \frac{\text{val}\mathcal{G}'(q \cdot s)}{q \cdot r}$.*

From now on we assume that every IPG we investigate is a flat game over \mathbb{Z}_∞ when not explicitly stated otherwise.

5 Discrete Strategies

Our goal in this section is to show that it suffices to use a simple kind of (almost) discrete strategies to approximate the value of flat interval parity games over \mathbb{Z}_∞ . To this end, we define an equivalence relation between states whose variables belong to the same \mathbb{Z} intervals. This equivalence, resembling the standard methods used to build the region graph from timed automata, is a technical tool needed to compare the values of the game in similar states.

Definition 16. *We say that two states s and t in an IPG are equivalent, $s \sim t$, if they are in the same location ($\text{loc}(s) = \text{loc}(t)$) and for all $i, j \in \{1, \dots, K\}$:*

- $[\text{var}_i(s)] = [\text{var}_i(t)]$, and
- if $\{\text{var}_i(s)\} \leq \{\text{var}_j(s)\}$ then $\{\text{var}_i(t)\} \leq \{\text{var}_j(t)\}$.

Intuitively, all variables lie in the same integer intervals and the order of fractional parts is preserved. In particular, it follows that all integer variables are equal. The following technical lemma allows to shift moves between \sim -states.

Lemma 17. *Let s and t be two states in a flat IPG over \mathbb{Z} such that $s \sim t$. If a move from s to s' is allowed by a label $l = (I, \overline{C}, R)$, then there exists a state t' , denoted $s'[s/t]$, the move to which from t is allowed by the same label l and*

- (1) $t' \sim s'$, and
- (2) there is no state $s'' \neq s'$ with (s, s'') allowed by l for which $s''[s/t] = t'$.

Using the lemma above, we can define the notion of shifting play histories. Let $h = t_0 t_1 \dots t_k$ be a play history such that (t_i, t_{i+1}) is allowed by label l_i and let s_0 be a state, $s_0 \sim t_0$. We say that $s_0 s_1 \dots s_k$ is a shifted history for h , from the view of player i , if the following conditions are satisfied. For every i if $t_i \in V_i$ then we require that $t_{i+1} = s_{i+1}[s_i/t_i]$. Note that if there is such a s_{i+1} then it is unique by condition (2) of the previous Lemma. If $t_i \in V_{1-i}$ then we require that $s_{i+1} = t_{i+1}[t_i/s_i]$. Note that if there exists a shifted history for h then it is uniquely determined by s_0 , and we will denote it (for player i) by $h^i[t_0/s_0]$.

Having defined shifted histories we can shift entire strategies. Given a strategy σ of player i and a state s_0 , we define

$$\sigma[s_0/t_0](t_0 \dots t_n) = \begin{cases} \sigma(s_0 \dots s_n)[s_n/t_n] & \text{if } s_0 \dots s_n = (t_0 \dots t_n)^i[s_0/t_0] \text{ exists,} \\ \sigma(t_0 \dots t_n) & \text{otherwise.} \end{cases}$$

This allows to shift whole strategies as stated below.

Lemma 18. *Let \mathcal{G} be a flat IPG over \mathbb{Z}_∞ and let $s_0 \sim t_0$ be two states of \mathcal{G} . For any strategies σ of Player 0 and ρ of Player 1 we consider the plays $\pi(\sigma, \rho^1[t_0/s_0], s_0) = s_0 s_1 \dots = \pi_s$ and $\pi(\sigma^0[s_0/t_0], \rho, t_0) = t_0 t_1 \dots = \pi_t$. It holds that either both π_s and π_t are infinite and $p(\pi_s) = p(\pi_t)$, or both are finite and of the same length $n+1$ and the last states of these plays s_n and t_n satisfy $s_n \sim t_n$.*

Using the property established above we can finally prove the following.

Lemma 19. *Let \mathcal{G} be a flat IPG over \mathbb{Z}_∞ with the maximal absolute value of the multiplicative factor in payoff functions m , and let $s_0 \sim t_0$. Then $|\text{val}\mathcal{G}(s_0) - \text{val}\mathcal{G}(t_0)| \leq m \cdot |s_0 - t_0|$.*

5.1 Choosing Discrete Moves

We show that for IPGs over \mathbb{Z}_∞ , fully general strategies are not necessary. In fact, we can restrict ourselves to discrete strategies and, using this, reduce the games to discrete systems. Intuitively, a discrete strategy keeps the maximal distance of all variable valuations to the closest integer small.

For the proof that there exist good discrete strategies it is convenient to work with the following notion of distance for a state. For $r \in \mathbb{R}$, define

$$d(r) = \begin{cases} r - \lceil r \rceil & \text{if } |r - \lceil r \rceil| \leq |r - \lfloor r \rfloor|; \\ r - \lfloor r \rfloor & \text{otherwise.} \end{cases}$$

This function gives the distance to the closest integer, except that it is negative if the closest integer is greater than r , i.e. if the fractional part of r is $> \frac{1}{2}$.

For a state s , we use the abbreviation $d_i(s) = d(\text{var}_i(s))$. We denote by $d_l(s) = \min_{i=1 \dots k} \{d_i(s)\}$ and $d_r(s) = \max_{i=1 \dots k} \{d_i(s)\}$ the smallest and biggest of all values $d_i(s)$, and additionally we define the total distance as follows.

$$d^*(s) = \begin{cases} |d_l(s)| & \text{if } d_i(s) \leq 0 \text{ for all } i \in \{1, \dots, k\}, \\ d_r(s) & \text{if } d_i(s) \geq 0 \text{ for all } i \in \{1, \dots, k\}, \\ |d_l(s)| + d_r(s) & \text{otherwise.} \end{cases}$$

First, we will prove that we can always correct a strategy that makes one step which is not ε -discrete. By doing so, we will guarantee that we reach a state with the same location that is allowed by the labelling and the values of the variables only change within the same intervals.

Lemma 20. *Let t be a state with $d^*(t) \leq \frac{1}{4}$ and s be a successor of t , where (t, s) is allowed by l . Then, for every $0 < \varepsilon < d^*(t)$, there exists a successor s'_+ of t such that $s \sim s'_+$, (t, s'_+) is allowed by l , and $d^*(s'_+) \leq d^*(t) + \varepsilon$.*

Knowing that in one step, the move can always preserve small total distance, we can finally define discrete strategies.

Definition 21. *We call a strategy σ ε -discrete if for every $s_{n+1} = \sigma(s_0 \dots s_n)$ it holds that if $d^*(s_n) \leq \varepsilon$ then $d^*(s_{n+1}) \leq d^*(s_n) + \frac{\varepsilon}{2}$.*

Observe that it follows directly from the definition that if $d^*(s_0) \leq \frac{\varepsilon}{2}$ and both players play discrete strategies, then $d^*(s_n) \leq \varepsilon(1 - \frac{1}{2^{n+1}})$.

Example 22. To see that decreasing ε in each step is sometimes crucial, consider the game with one variable depicted in Figure 4. In each move Player 0 has to choose a positive value in $(0, 1)$. Player 1 can then decide to continue the play or leave the cycle and end the play with the negative accumulated value, i.e. $-x_0$, as payoff. He cannot infinitely often decide to stay in the cycle as then the payoff would be ∞ as the priority is 0. An ε -optimal strategy for Player 0 as the maximising player is thus to start with $\frac{\varepsilon}{2}$ and decrease in each step.

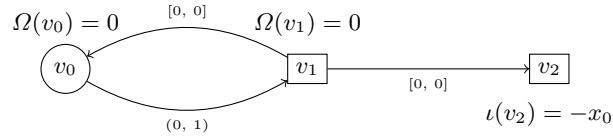


Fig. 4. Game in which the values played must decrease.

We now extend the previous lemma to one that allows to shift a whole move.

Lemma 23. *Let s be a state and s' a successor of s , where (s, s') is allowed by l . Let t be a state with $d^*(t) \leq \frac{1}{4}$, such that $s \sim t$. Then for every $\varepsilon > 0$ there exists a successor t' of t allowed by l such that $t \sim t'$ and $d^*(t') \leq d^*(t) + \varepsilon$.*

We can conclude that discrete strategies allow to approximate game values.

Lemma 24. *For every strategy σ of Player i in \mathcal{G} , there is a discrete strategy σ_d , such that for any starting state s_0 and discrete strategy ρ of the other player, if $\pi(\sigma, \rho, s_0) = s_0, s_1, \dots$ and $\pi(\sigma_d, \rho, s_0) = s'_0, s'_1, \dots$, then $s_i \sim s'_i$ for all i .*

Proposition 25. *Let \mathcal{G} be a flat interval parity game. Let Γ_i be the set of all strategies for player i and Δ_i the set of all discrete strategies for player i and m be the highest value that occurs as a multiplicative factor in ι . Then it holds, for every starting state s , that*

$$|\sup_{\sigma \in \Gamma_0} \inf_{\rho \in \Gamma_1} p(\pi(\sigma, \rho, s)) - \sup_{\sigma \in \Delta_0} \inf_{\rho \in \Delta_1} p(\pi(\sigma, \rho, s))| \leq m.$$

6 Counter-Reset Games

By the above Proposition 25, we can restrict both players to use ε -discrete strategies to approximate the value of a flat interval game up to the maximal multiplicative factor m . Multiplying the game by any number q does not change the multiplicative factors in ι but multiplies the value of the game by q . Thus, to approximate the value of \mathcal{G} up to $\frac{1}{n}$ it suffices to play ε -discrete strategies in $n \cdot m \cdot \mathcal{G}$. When players use only discrete strategies, the chosen values remain close to integers (possibly being up to ε bigger or smaller). The fact whether the value is bigger, equal or smaller than an integer can be stored in the state, as well as whether the value of a variable is smaller than any of the (non-infinite) bounds in constraint intervals or bigger than all of them. This way, we can eliminate both ε 's and constraints and are left with the following games.

Definition 26. *A counter-reset game is a flat interval parity game in which in each label $l = (I, \bar{C}, R)$ the constraints \bar{C} are trivially true and the interval I is either $[0, 0]$ or $[1, 1]$, i.e. either all variables are incremented by 1 or all are left intact. A generalised counter-reset game is one in which each variable separately is assigned to be incremented or to be left intact in each move.*

Lemma 27. *Let \mathcal{G} be an IPG over \mathbb{Z}_∞ with maximal absolute value of the multiplicative factor in ι equal to m . For each $n \in \mathbb{N}$ there exists a counter-reset game \mathcal{G}'_n such that for all states s in which all variables are integers:*

$$|\text{val}\mathcal{G}(s) - \frac{\text{val}\mathcal{G}'_n(n \cdot m \cdot s)}{n \cdot m}| \leq \frac{1}{n}.$$

We solve (even the generalised) counter-reset games in a similar way as classical parity games. We start with games of finite duration and observe that there is a simple symbolic representation for the payoffs, in terms of min-max functions, which can be achieved by the players. Then, we use the unfolding theorem from [6] and compute fixed-points over this representation to solve these parity games. To exploit this fixed-point computation algorithmically, it is necessary to show that a fixed-point in the chosen symbolic representation will be reached in a finite number of steps. To this end, we use a form of Dickson's Lemma applied to linearly controlled functions [10, 5]. This allows us to show convergence of the fixed-points and thus exactly calculate the value of a counter-reset game.

Proposition 28. *Given a generalised counter-reset game \mathcal{G} and a state s in which all counters are integers, one can compute $\text{val}\mathcal{G}(s)$.*

7 Conclusions and Future Work

We conclude by completing the proof of our main Theorem 7. We first observe that, by Theorem 10, evaluating a $Q\mu$ -formula on a system is equivalent to calculating the value of the corresponding model-checking game. We can then turn this game into a flat one by Lemma 12 and then into one over \mathbb{Z}_∞ by Corollary 15. By Lemma 27 the value of such a game can be approximated arbitrarily precise by counter-reset games, which we can solve by Proposition 28.

All together, we proved that it is possible to approximate the values of quantitative μ -calculus formulae on initialised linear hybrid systems with arbitrary precision. Unfortunately, we cannot give complexity bounds for our procedure, even though we believe that the algorithm we presented is elementary (and related to questions about cost-MSO, a recently studied logic [2]). Two immediate problems remain open: (1) can the exact value of $\llbracket \varphi \rrbracket^{\mathcal{K}}$ be computed? (2) what is the complexity of such a computation or its approximation? Even with further research needed to answer these questions, our result lays the foundation for using temporal logics for the quantitative verification of hybrid systems.

References

1. Rajeev Alur, Costas Courcoubetis, Nicolas Halbwachs, Thomas A. Henzinger, Pei-Hsin Ho, Xavier Nicollin, Alfredo Olivero, Joseph Sifakis, and Sergio Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138(1):3–34, 1995.
2. Thomas Colcombet and Christof Löding. Regular cost functions over finite trees. In *LICS*, pages 70–79. IEEE Computer Society, 2010.
3. Luca de Alfaro. Quantitative verification and control via the mu-calculus. In *CONCUR*, volume 2761 of *LNCS*, pages 102–126. Springer, 2003.
4. Luca de Alfaro, Marco Faella, and Mariëlle Stoelinga. Linear and branching metrics for quantitative transition systems. In *ICALP*, volume 3142 of *LNCS*, pages 97–109. Springer, 2004.
5. Diego Figueira, Santiago Figueira, Sylvain Schmitz, and Philippe Schnoebelen. Ackermann and primitive-recursive bounds with dickson’s lemma. *CoRR*, abs/1007.2989, 2010.
6. Diana Fischer, Erich Grädel, and Łukasz Kaiser. Model checking games for the quantitative μ -calculus. *Theory Comput. Syst.*, 47(3):696–719, 2010.
7. Thomas Gawlitza and Helmut Seidl. Computing game values for crash games. In *ATVA*, volume 4762 of *LNCS*, pages 177–191. Springer, 2007.
8. Thomas A. Henzinger, Benjamin Horowitz, and Rupak Majumdar. Rectangular hybrid games. In *Proceedings of CONCUR’99*, volume 1664 of *LNCS*, pages 320–335. Springer, 1999.
9. Thomas A. Henzinger, Peter W. Kopke, Anuj Puri, and Pravin Varaiya. What’s decidable about hybrid automata? In *Proceedings of STOC’95*, pages 373–382. ACM, 1995.
10. Ken McAloon. Petri nets and large finite sets. *Theoretical Computer Science*, 32:173–183, 1984.
11. Annabelle McIver and Carroll Morgan. Results on the quantitative μ -calculus $qM\mu$. *ACM Trans. Comput. Log.*, 8(1), 2007.