
MODEL CHECKING THE QUANTITATIVE μ -CALCULUS ON LINEAR HYBRID SYSTEMS

DIANA FISCHER^a AND LUKASZ KAISER^b

^a Mathematische Grundlagen der Informatik, RWTH Aachen University
e-mail address: fischer@logic.rwth-aachen.de

^b LIAFA, CNRS & Université Paris Diderot – Paris 7
e-mail address: kaiser@liafa.univ-paris-diderot.fr

ABSTRACT. We study 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 that arise naturally in this context. Recently, quantitative formalisms that subsume classical temporal logics and allow the measurement of 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 present a reduction to a class of counter parity games.

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, 14, 15], it is also interesting to infer quantitative values for certain queries. For example, one may not only want to check that a variable of a system does not exceed a given threshold, but also to compute the maximum value of the variable over all runs, checking whether any such threshold exists.

Thus far, quantitative testing of hybrid systems has only been done by simulation, and hence lacks the strong guarantees which can be given by model checking. In recent years, there has been a strong interest in extending classical model-checking techniques and logics to the quantitative setting. Several quantitative temporal logics have been introduced, see e.g. [5, 6, 7, 8, 10, 11, 17], 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

1998 ACM Subject Classification: D.2.4, F.4.1.

Key words and phrases: hybrid systems, model checking, μ -calculus, quantitative logics, games.

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

algorithm for a non-stochastic quantitative temporal logic on a class of hybrid systems. The logic we consider, the quantitative μ -calculus [8], is based on a formalism first introduced in [6]. It properly subsumes the standard μ -calculus, cf. [4], and 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 [14, 15], which is one of the strongest model-checking results for hybrid systems.

The restriction to initialised linear systems is made because verification of temporal properties over general hybrid systems is undecidable. This holds even for linear systems, thus one must pick an appropriate abstraction of the system. An established and very well-studied way to do this is to first approximate the continuous behaviour of the variables by linear behaviour in a finite number of intervals. This method, applied to a number of functions $f_1(x), \dots, f_m(x)$ that evolve according to a set of arbitrary differential equations $\mathcal{D}(f_1, \dots, f_m)$, generates a set of disjoint intervals I_1, \dots, I_k with $I_1 \cup \dots \cup I_k = \mathbb{R}$ and a set of linear coefficients a_i^j, b_i^j such that in I_j it is approximately true that $f_i(x) = a_i^j \cdot x + b_i^j$, i.e. the derivative $\frac{df_i}{dt} = a_i^j$. There are several ways to generate such linear approximations of solutions of differential equations and, depending on the method in question, one can obtain various kinds of error bounds for the respective classes of functions. We do not investigate these issues (or other approximation methods) here, but focus instead on the linear system obtained.

As stated above, even simple qualitative verification problems are undecidable for general hybrid systems. This remains true even after the natural approximation by a linear system. Hence, one more assumption is made, namely that if the speed of evolution of a variable changes between discrete locations then also the variable is reset on that transition. Systems with this property, called *initialised* linear systems, are – besides o-minimal systems [16, 3] and their recent extensions [18] – one of the largest classes of hybrid systems with decidable temporal logic [15]. Observe that when an arbitrary hybrid system is approximated by a linear one, one can *try* to directly obtain an initialised system by computing boundary values [13]. This can be done by either assuring that discrete transitions are taken only at the borders of the intervals I_j , or by taking a finer subdivision of the intervals to increase the precision of coordination between the discrete and the continuous part of the system. Note that, even though this procedure has been implemented in model-checking programs, it is only a heuristic – it necessarily fails for general systems for which the model-checking problem is undecidable.

The logic we study is quantitative – it 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 working directly with a quantitative equivalence relation, roughly similar to the region graph for timed automata, and finally by exploiting a recent result on counter parity games.

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 reduce the problem to counter parity games and exploit a recent result to solve them. To sum up, the steps taken are depicted below.

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

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 $\mathbb{Z}_\infty, \mathbb{Q}_\infty$ and \mathbb{R}_∞ .

Definition 2.1. 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. The set \mathcal{L}_M of *transition labels* consists of triples $l = (I, \overline{C}, R)$, where the vector $\overline{C} = (C_1, \dots, C_M)$ (with $C_i \in \mathcal{I}(\mathbb{R}_\infty)$ for $i \in \{1, \dots, M\}$) represents the constraints each of the variables needs 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 y_i is set to zero. 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 y_i the coefficient a_i such that the variable evolves in this location according to the equation $\frac{dy_i}{dt} = a_i$.

Please note that although we do not explicitly have any invariants (or constraints) in locations, we can simulate them by choosing either the time intervals or variable constraints on the outgoing transitions accordingly. If the values of predicates and labels range over \mathbb{Q}_∞ or \mathbb{Z}_∞ instead of \mathbb{R}_∞ , we talk about linear hybrid systems over \mathbb{Q} and \mathbb{Z} , respectively.

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 $(I, \overline{C}, R) \in \lambda((v, v'))$ if $\overline{y} \in \overline{C}$ (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, \overline{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 $s = (v, \overline{y})$ and $s' = (v', \overline{y}')$.

Definition 2.2. A linear hybrid system \mathcal{K} is *initialised* if for each $(v, w) \in E$ and each variable y_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.

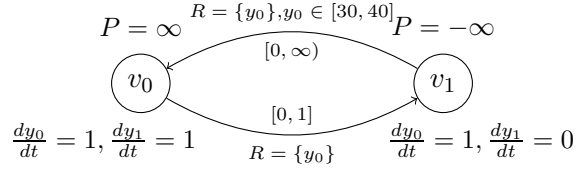


Figure 1: Leaking gas burner LHS $\mathcal{L} = (V, E, P, \lambda, \delta)$ (not initialised)

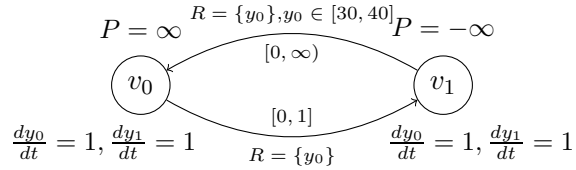


Figure 2: Leaking gas burner LHS $\mathcal{L} = (V, E, P, \lambda, \delta)$ (initialised)

Example 2.3. To clarify the notions we use, we consider a variant of a standard example for a linear hybrid system, the leaking gas burner.

Our version is depicted in Figure 1. This system represents a gas valve that can leak gas to a burner, so it has two states: v_0 , where the valve is open (and leaking gas) and v_1 where it is closed. This is also indicated by a qualitative predicate P that has the value ∞ if the gas is leaking (in location v_0) and $-\infty$ otherwise. The system has two variables. The first variable, y_0 , is a clock measuring the time spent in each location, and is reset on each transition, i.e. after each discrete system change. The variable y_1 is a stop watch and measures the total time spent in the leaking location. Thus, this system is not initialised. The time intervals on the transitions control the behaviour of the system. On the transition (v_0, v_1) there are no restrictions on the variables, but we are only allowed to choose a time unit from $[0, 1]$, i.e. we can stay a maximum of one time unit in location v_0 . On the transition (v_1, v_0) there is a restriction on the value of y_0 , it has to have a value between 30 and 40 for this transition to be allowed, while there is no restriction on the choice for the time unit (of course, this could also be modelled the other way around). Intuitively, the time intervals indicate that the gas valve will leak gas for a time interval between 0 and 1 seconds and then be stopped and that it can only leak again after at least 30 time units.

In Figure 2, we show an initialised version of the leaking gas burner. The only difference is that y_1 is not a stop watch anymore but a normal clock. Since now both variables are just clocks (which means that their evolution rates are one everywhere), the system is trivially initialised.

2.1. Quantitative μ -Calculus. In this section, we present a version of the quantitative μ -calculus first introduced in [8]. The version we use here is additive and includes variables. It is evaluated on linear hybrid systems.

Definition 2.4. Given sets of fixpoint variables \mathcal{X} , system variables $\{y_1, \dots, y_M\}$ and predicates $\{P_i\}_{i \in J}$, the formulae of the *quantitative μ -calculus ($Q\mu$) with variables* are given

by the EBNF grammar:

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

where $X_j \in \mathcal{X}, y_k \in \{y_1, \dots, y_M\}$, and in the cases $\mu X_j.\varphi$ and $\nu X_j.\varphi$, the variable X_j must appear positively in φ , i.e. under an even number of negations.

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

Definition 2.5. Given a linear hybrid system $\mathcal{K} = (V, E, \lambda, \{P_i\}_{i \in J}, \delta)$ and an interpretation \mathfrak{J} , a $Q\mu$ -formula yields a valuation function $\llbracket \varphi \rrbracket_{\mathfrak{J}}^{\mathcal{K}} : S \rightarrow \mathbb{R}_\infty$ defined in the following standard way for a state $s = (v^s, y_1^s, \dots, y_M^s)$.

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

For formulae without free variables we write $\llbracket \varphi \rrbracket^{\mathcal{K}}$ rather than $\llbracket \varphi \rrbracket_{\mathfrak{J}}^{\mathcal{K}}$.

Please note that the inclusion of variables does not fundamentally change the semantics of quantitative μ -calculus. The quantitative μ -calculus in [8] is evaluated on quantitative transition systems. Here, a formula is evaluated on the state graph of a linear hybrid system, rather than the system itself. Intuitively, a linear hybrid system is a compact representation of an infinite quantitative transition system (its state graph). Thus, many properties of the quantitative μ -calculus from [8] remain true. For example, to embed the classical μ -calculus in quantitative μ -calculus one must interpret *true* as $+\infty$ and *false* as $-\infty$.

Example 2.6. The formula $\mu X.(\Diamond X \vee y_1)$ evaluates to the supremum of the values of y_1 on all runs from some initial state: e.g. to ∞ if evaluated on the simple initialised leaking gas burner model. To determine the longest period of time during which the gas is leaking we use the formula $\mu X.(\Diamond X \vee (y_0 \wedge P))$, which evaluates to 1 on the initial state $(v_0, \bar{0})$ in our example.

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

In other words, for every ε we can approximate $\llbracket \varphi \rrbracket^{\mathcal{K}}$ within ε . We formulated the theorem above using n because it makes the representation of ε precise, so we can provide a complexity bound: Given on input the system \mathcal{K} , the formula φ and n , we will show how to compute the number r (or output $\pm\infty$) in 8EXPTIME.

3. INTERVAL GAMES

In this section, we define a variant of quantitative parity games suited for model checking $Q\mu$ on linear hybrid systems. As mentioned above, a linear hybrid system can be seen as a compact representation of an infinite quantitative transition system. Similarly, a parity game that is played on a linear hybrid system can be viewed as a compact, finite description of an infinite quantitative parity game, as defined in [8].

Definition 3.1. 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) (without predicates) 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 $\lambda : E \rightarrow \mathcal{P}_{\text{fin}}(\mathcal{L}_M)$. 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.

Please note that interval parity games are played on linear hybrid systems without any quantitative predicates, i.e. the set of predicates is empty and therefore omitted.

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 $S = S_0 \dot{\cup} S_1$.

How to play. Every play starts at some position $v \in V$ with all variables set to 0, i.e. the starting state is $s_0 = (v, 0, \dots, 0)$. For every state $s = (v, \bar{y}) \in S_i$, player i chooses an allowed successor state $s' \in \text{succ}(s)$ and the play proceeds from s' . If the play reaches a state s such that $\text{succ}(s) = \emptyset$ it ends, otherwise the play is infinite.

Intuitively, the players choose the time period they want to spend in a location before taking a specified transition. Note that in this game every position could possibly be a terminal position. This is the case if it is not possible to choose a time period from the given intervals in such a way that the respective constraints on all variables are fulfilled.

Payoffs. The outcome $p(s_0 \dots s_k)$ of a finite play ending in $s_k = (v, y_1, \dots, y_M)$ where $\iota(v) = (i, a, b)$ is $p(s_k) = a \cdot y_i + b$. To improve readability, from now on we will simply write $\iota(v) = a \cdot y_i + b$ in this case. The outcome of an infinite play depends only on the lowest priority seen infinitely often in positions of the play. We will assign the value $-\infty$ to every infinite play, where the lowest priority seen infinitely often is odd, and ∞ to those where it is even.

Goals. The two players have opposing objectives regarding the outcome of the play. Player 0 wants to maximise the outcome, while Player 1 wants to minimise it.

Strategies. A strategy for player $i \in 0, 1$ is a function $\sigma : S^* S_i \rightarrow S$ with $\sigma(s) \in \text{succ}(s)$. A play $\pi = s_0 s_1 \dots$ is *consistent with a strategy* σ for player i , if $s_{n+1} = \sigma(s_0 \dots s_n)$ for every n such that $s_n \in S_i$. For strategies σ, ρ for the two players, we denote by $\pi(\sigma, \rho, s)$ the unique play starting in state s which is consistent with both σ and ρ .

Determinacy. A game is *determined* if, for each state s , the highest outcome Player 0 can assure from this state and the lowest outcome Player 1 can assure coincide,

$$\sup_{\sigma \in \Gamma_0} \inf_{\rho \in \Gamma_1} p(\pi(\sigma, \rho, s)) = \inf_{\rho \in \Gamma_1} \sup_{\sigma \in \Gamma_0} p(\pi(\sigma, \rho, s)) =: \text{val}\mathcal{G}(s),$$

where Γ_0, Γ_1 are the sets of all possible strategies for Player 0, Player 1 and the achieved outcome is called the *value of \mathcal{G} at s* .

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.

Intuitively, in a play of an interval parity game, the players choose successors of the current state as long as possible.

Example 3.2. In Figure 3, we show a simple example of an interval parity game. Positions of Player 0 are depicted as circles and positions of Player 1 as boxes. To keep things simple, there is just one clock variable, y_0 , all constraints are trivially true and the reset sets are empty, so we label the transitions only with the time intervals that the players can choose from. The priorities are depicted next to the nodes for non-terminal positions and the evaluation function above the terminal position (in general, also positions with outgoing edges could be terminal, however in this example this is not possible as there are no constraints on the variable).

A play of this system starting at node v_0 could end after two moves in position v_2 , if Player 1 decided to move there (he also has the choice to move down). The payoff of this play would then depend only on the choice that Player 0 made in the first move, for example $\frac{1}{3} \in [0, \frac{1}{2}]$. Then the payoff would be $3 \cdot (\frac{1}{3} + 2) - 1 = 6$ (as in this play, the second time interval only permits the choice 2).

If Player 1 would move down instead of ending the play and the play would loop infinitely often in the cycle v_3, v_4, v_5 at the bottom, the least priority that occurs infinitely often would determine the outcome of the play; in this case it would be 0 at v_3 and therefore the payoff would be ∞ .

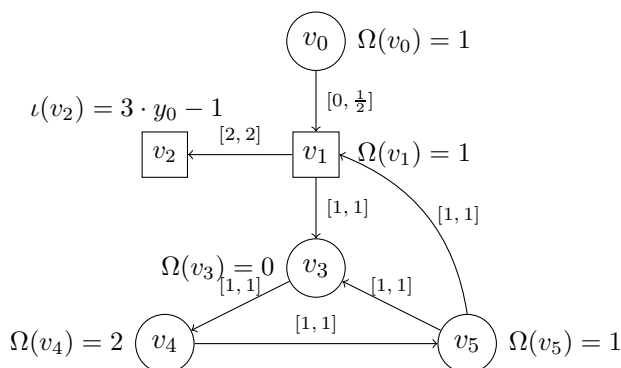


Figure 3: Simple interval parity game

We already mentioned that an interval parity game can be seen as a representation of a quantitative parity game, now we want to describe this formally. We use the notion from [8] 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 $\mathcal{G}^* = (V_0 \times \mathbb{R}_\infty^M, V_1 \times \mathbb{R}_\infty^M, E^*, \lambda^*, \Omega^*)$

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 quantitative parity game \mathcal{G}^* in [8]. In particular, it follows from the determinacy of quantitative parity games that also interval parity games are determined.

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 [8] and is presented below.

Intuitively, the positions are pairs consisting of a subformula of φ and a location of \mathcal{K} . Which player moves at which position depends on the outermost operator of the subformula. At disjunctions Player 0 moves 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.

Definition 3.3. For a linear hybrid system $\mathcal{K} = (V, E, \{P_i\}_{i \in J}, \lambda, \delta)$ and a $Q\mu$ -formula φ in negation normal form, the interval game

$$\text{MC}[\mathcal{K}, \varphi] = (V_0, V_1, E, \lambda, \delta, \iota, \Omega),$$

which we call the *model-checking game* for \mathcal{K} and φ , is constructed in the following way, similar to the standard construction of model-checking games for the μ -calculus (c.f. [8]).

Positions. The positions of the game are pairs (ψ, v) , where ψ is a subformula of φ , and $v \in V$ is a location in the LHS \mathcal{K} . Positions (ψ, v) where the top operator of ψ is \square, \wedge , or ν belong to Player 1 and all other positions belong to Player 0. A state in the game is denoted by $s = (p, \bar{y})$, where $p = (\psi, v)$ is the position and \bar{y} is the variable assignment of the location v in the underlying linear hybrid system \mathcal{K} .

Moves. Positions of the form (P_i, v) and (y_i, v) are terminal positions. From positions of the form $(\psi \wedge \theta, v)$, resp. $(\psi \vee \theta, v)$, one can move to (ψ, v) or to (θ, v) . Positions of the form $(\diamond\psi, v)$ have either a single successor $(-\infty)$ in case v is a terminal location in \mathcal{K} , or one successor (ψ, v') for every $v' \in vE$. Analogously, positions of the form $(\square\psi, v)$ have a single successor (∞) if $vE = \emptyset$, or one successor (ψ, v') for every $v' \in vE$ otherwise. The moves corresponding to system moves (v, v') are labelled accordingly with $\lambda((v, v'))$, all other moves are labelled with the empty label $([0, 0], (-\infty, \infty)^M, \emptyset)$ which indicates that no time passes, there are no constraints on the variables and no variable is reset. Fixed-point positions $(\mu X.\psi, v)$, resp. $(\nu X.\psi, v)$ have a single successor (ψ, v) . Whenever one encounters a position where the fixed-point variable stands alone, i.e. (X, v') , the play goes back to the corresponding definition, to (ψ, v') .

Payoffs. The function ι assigns $\llbracket P_i \rrbracket(v)$ to all positions (P_i, v) , $\pm\infty$ to all positions $(\pm\infty)$ and y_i to positions (y_i, v) . To discourage the players from ending the game at any other position than a terminal one, ι assigns all other positions outcome $-\infty$ for Player 0's

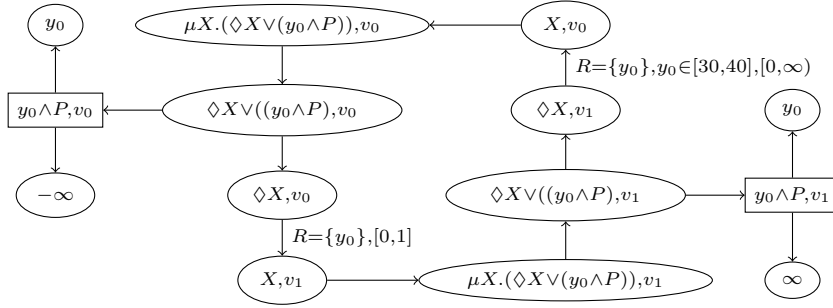


Figure 4: Model-checking game for $\mu X.(\diamond X \vee (y_0 \wedge P))$ on initialised leaking gas burner.

positions or ∞ for Player 1's positions. The payoff $p(\pi)$ of a play π is calculated using ι and the priorities as stated before.

Priorities. The priority function Ω is defined as in the classical case using the alternation level of the fixed-point variables, see e.g. [12]. Positions (X, v) get a lower priority than positions (X', v') if X has a lower alternation level than X' . The priorities are then adjusted to have the right parity, such that an even value is assigned to all positions (X, v) where X is a ν -variable and an odd value to those where X is a μ -variable. The maximum priority, equal to the alternation depth of the formula, is assigned to all other positions.

Example 3.4. We continue our example of the leaking gas burner and present in Figure 4 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. There is only one position with a constraints on variable y_0 and in only two positions a choice about the time that passes can be made. Both of these positions belong to Player 0 in this example and are labelled with the corresponding intervals below (and in both y_0 is also reset). In terminal nodes, either the variable y_0 or the predicate P is evaluated for the payoff (this choice can be made by Player 1 in this example). 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 y_0 or P at v_1 . Since he is minimising, he will choose to evaluate y_0 .

It has been shown in [8] 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 3.5. *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*

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

Proof. Determinacy of an interval parity game \mathcal{G} follows directly from the determinacy of the infinite QPG \mathcal{G}^* used to define \mathcal{G} .

Let φ be a $Q\mu$ -formula and \mathcal{K} a linear hybrid system. Let $S(\mathcal{K}) = (S, E^S)$ be the state graph of \mathcal{K} , where S is the set of all states, and $(s, s') \in E^S$ iff $s' \in \text{succ}(s)$ in \mathcal{K} . Let $\mathcal{K}^* = (S, E^S, P_{y_0} \dots P_{y_M})$ be the quantitative transition system with predicates P_{y_i} where

$P_{y_i}(v, \bar{a}) = a_i$. Let us also rewrite the formula φ into a formula without variables, φ^* , by replacing each occurrence of y_i by the corresponding P_{y_i} .

Applying the model-checking Theorem 12 from [8] we conclude that for all $v \in \mathcal{K}^*$ it holds $\text{valMC}[\mathcal{K}, \varphi]^*(\varphi, v) = \llbracket \varphi \rrbracket^{*\mathcal{K}^*}(v)$, i.e. that $\text{MC}[\mathcal{K}, \varphi]^*$ is the model-checking game for \mathcal{K}^* and φ^* . Finally, by definition of IPGs on the one hand and the semantics of $Q\mu$ on the other, it follows that for all \bar{x}

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

4. BASIC PROPERTIES OF INTERVAL GAMES

In this section, we first give a brief example that illustrates the difference between interval games and timed games. Then, we show how to transform an initialised interval game over \mathbb{Q}_∞ into an easier game over \mathbb{Z}_∞ in which the all evolution rates are one.

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 discretised. To stress that quantitative payoffs indeed make a difference, we present in Figure 5 an initialised interval parity game with the interesting property that it is not optimal 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 this variable 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 (they are chosen to be unfavourable for the current player such that she has to continue playing). The payoff rule specifies the outcome of a play π ending in v_2 as $p(\pi) = y_0 - 1$ and in v_3 as $p(\pi) = -y_0$. 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.

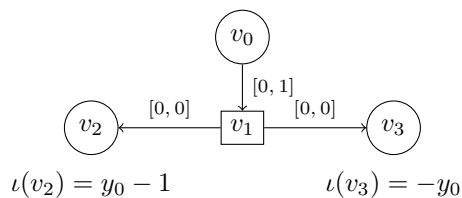


Figure 5: Game with integer coefficients and non-integer value.

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.

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.

Definition 4.1. An interval parity game $\mathcal{G} = (V_0, V_1, E, \lambda, \delta, \iota, \Omega)$ is *flat* if and only if $\delta_i(v) = 1$ for all $v \in V$ and $i = 1 \dots M$.

Lemma 4.2. For each initialised interval parity game \mathcal{G} there exists a flat game \mathcal{G}' with the same value.

Proof. Let $\mathcal{G} = (V_0, V_1, E, \lambda, \delta, \iota, \Omega)$ be an initialised interval parity game. We construct a corresponding flat game $\mathcal{G}' = (V_0, V_1, E, \lambda', \delta', \iota', \Omega)$ in the following way: For a position $v \in V = V_0 \dot{\cup} V_1$ and each variable y_i , such that $\delta_i(v) = a_i$, $\iota(v) = a \cdot y_i + b$ and an outgoing edge (v, w) with $C_i = [c_0, c_1]$ we have in the corresponding flat game:

- $\delta'_i(v) = 1$
- $C'_i \in \lambda'(v, w) = [\frac{c_0}{a_i}, \frac{c_1}{a_i}] = \frac{1}{a_i} C_i$
- $\iota'(v) = a_i \cdot a \cdot y_i + b$

Note that we only change the functions δ, λ and ι . We will show that for every play π from a starting state s consistent with σ and ρ , we can construct strategies σ', ρ' , such that $\pi'(\sigma', \rho', s')$ visits the same locations as π and $p(\pi) = p(\pi')$. Before we proceed with the proof, notice that it is essential that \mathcal{G} is an initialised game. Intuitively, the value of y_i in \mathcal{G}' is the value of y_i in \mathcal{G} divided by the coefficient a_i of the current position. When the position changes, it is thus crucial that a_i does *not* change, except if y_i is reset – exactly what is required from an initialised game.

The proof proceeds by induction on the length of the plays. First, if $s_0 = (v_0, \bar{0})$ is a state belonging to Player 0 and $\sigma(s_0) = s_1 = (v_1, \bar{x})$ and $s'_0 = (v_0, \bar{0})$, then in \mathcal{G}' we define $\sigma'(s'_0) = s'_1$, where $s'_1 = (v_1, \bar{y}')$, such that $y'_i = \frac{y_i}{a_i}$ for any $y_i \notin R \in \lambda(v_0, v_1)$. Since (s_0, s_1) is allowed in \mathcal{G} , this means that for all $y_i \notin R \in \lambda(v_0, v_1)$, we have $y_i \in C_i = [c_0, c_1] \in \lambda(v_0, v_1)$. It follows that $\frac{c_0}{a_i} \leq y'_i = \frac{y_i}{a_i} \leq \frac{c_1}{a_i}$ for all $y_i \notin R$ and therefore (s'_0, s'_1) is allowed in \mathcal{G}' . Also $p(s_1) = \iota(v_1) = a \cdot y_i + b$ and therefore the payoff is equal to $p(s'_1) = \iota'(s'_1) = a_i \cdot a \cdot \frac{y_i}{a_i} + b$.

Let $s_0 \dots s_k$ and $s'_0 \dots s'_k$ be finite histories in \mathcal{G} and \mathcal{G}' , such that they visit the same locations and $p(\pi) = p(\pi')$. Then, if $s_k = (v_k, \bar{y})$ is a state belonging to Player 0 and $\sigma(s_k) = s_{k+1} = (v_{k+1}, \bar{y})$ and $s'_k = (v_k, \bar{z})$, then in \mathcal{G}' we define $\sigma'(s'_k) = s'_{k+1}$, where $s'_{k+1} = (v_{k+1}, \bar{w})$, such that $w_i = t$ where $t_i = \frac{y_i}{a_i}$ for any $y_i \notin R \in \lambda(v_k, v_{k+1})$. Since (s_k, s_{k+1}) is allowed in \mathcal{G} , this means that for all $y_i \notin R$, $y_i \in C_i = [c_0, c_1] \in \lambda(v_k, v_{k+1})$. As $\frac{c_0}{a_i} \leq w_i = \frac{y_i}{a_i} \leq \frac{c_1}{a_i}$ for all $y_i \notin R$, we get that (s'_k, s'_{k+1}) is allowed in \mathcal{G}' . Also $p(s_k) = \iota(v_k) = a \cdot y_i + b$ and therefore the payoff is equal to $p(s'_{k+1}) = \iota'(s'_{k+1}) = a_i \cdot a \cdot w_i + b = a_i \cdot a \cdot \frac{y_i}{a_i} + b$.

The cases for Player 1 are analogous. Note that, for infinite plays, we also have the same payoff, since for the payoff of infinite games only the locations (and their priorities) matter. Since we can construct, for each pair of strategies in \mathcal{G} , the corresponding strategies in \mathcal{G}' , and those yield a play with the same payoff, the values of the two games are equal. \square

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 4.3. 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 $\iota'(v) = a \cdot y_i + q \cdot b$ iff $\iota(v) = a \cdot y_i + b$ 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), and 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 4.4. *For every IPG \mathcal{G} over \mathbb{Q}_∞ and $q \in \mathbb{Q}, q \neq 0$ it holds in all states s that $q \cdot \text{val}\mathcal{G}(s) = \text{val } q \cdot \mathcal{G}(q \cdot s)$.*

Proof. We denote by $q \cdot \sigma$ the strategy with $q \cdot \sigma(q \cdot h) = q \cdot s'$ iff $\sigma(h) = s'$. The mapping of \mathcal{G} with strategies for both players σ and ρ to $q \cdot \mathcal{G}$ with $q \cdot \sigma$ and $q \cdot \rho$ is a bijection (in the reverse direction take $\frac{1}{q}$). We also have $q \cdot p_{\mathcal{G}}(\pi(\sigma, \rho, s) = s_0 s_1 \dots s_k) = q \cdot (a \cdot y_i + b)$ where $\iota(\text{loc}(s_k)) = (a, i, b)$ which is equal to $p_{q \cdot \mathcal{G}}(\pi(q \cdot \sigma, q \cdot \rho, q \cdot s) = q \cdot s_0 \dots q \cdot s_k) = a \cdot (q \cdot y_i) + q \cdot b$ for all finite plays π . Therefore, we know that $\inf_{\rho} q \cdot p(\pi(\sigma, \rho, s) = \inf_{q \cdot \rho} p(\pi(q \cdot \sigma, q \cdot \rho, q \cdot s)$ and the same holds for the supremum and thus we get the desired result. \square

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 4.5. *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.

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 5.1. 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 $i, j \in \{1, \dots, K\}$:

- $\lfloor \text{var}_i(s) \rfloor = \lfloor \text{var}_i(t) \rfloor$, 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 for the shifting of moves between \sim -states.

Lemma 5.2. *Let s and s' be two states in a flat IPG over \mathbb{Z} such that $s \sim s'$. If a move from s to t is allowed by a label $l = (I, \overline{C}, R)$, then there exists a state t' , the move to which from s' is allowed by the same label l and $t' \sim t$.*

Proof. If $R = \{1, \dots, K\}$ then let $t' = t$. As $s \sim s'$, the same constraints are satisfied by s and s' and thus the move from s' to $t' = t$ is allowed by the same label.

If $R \neq \{1, \dots, K\}$ then let $w = t -_R s \in I$ be the increment chosen during the move. If $w \in \mathbb{Z}$ we let $t' = s' + w$, the conditions follow from the assumption that $s \sim s'$ again.

If $w \notin \mathbb{Z}$, let i be the index of a non-reset variable with the smallest fractional part in t , i.e. $\{\text{var}_i(t)\} \leq \{\text{var}_j(t)\}$ for all $j \notin R$. To construct t' , we must choose w' with $\lfloor w' \rfloor = \lfloor w \rfloor$ which makes $\text{var}_i(s' + w')$ the one with smallest fractional part.

Case 1: $\{\text{var}_i(t)\} \geq \{w\}$.

In this case, for all non-reset variables j , holds $\{\text{var}_j(t)\} \geq \{w\}$, intuitively meaning that no variable “jumped” above an integer due to $\{w\}$. Let l be the variable with maximum fractional part in s' (and thus, by definition of \sim , also in s and in this case in t). Set

$$w' = \lfloor w \rfloor + 0.9 \cdot (\lceil \text{var}_l(s') \rceil - \text{var}_l(s')).$$

Clearly $\lfloor w' \rfloor = \lfloor w \rfloor$ and indeed, we preserved the order of fractional parts and integer intervals, thus \sim is preserved.

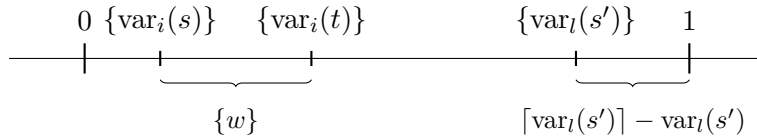


Figure 6: Lemma 5.2 Case 1

Case 2: $\{\text{var}_i(t)\} < \{w\}$ and for all $j \notin R$ $\{\text{var}_j(s')\} \geq \{\text{var}_i(s')\}$.

In this case, for all non-reset variables j , holds $\{\text{var}_j(t)\} \leq \{w\}$, intuitively meaning that all variables “jumped” above an integer due to $\{w\}$. Let l be the variable with maximum fractional part in s' (and thus also in s). Let

$$\delta = 0.9 \cdot \min(\{\text{var}_i(s')\}, (\lceil \text{var}_l(s') \rceil - \text{var}_l(s')))$$

be a number smaller than both $\{\text{var}_i(s')\}$ and $\lceil \text{var}_l(s') \rceil - \text{var}_l(s')$. We set

$$w' = \lfloor w \rfloor + \lceil \text{var}_i(s') \rceil - \text{var}_i(s') + \delta.$$

By the first assumption on δ we have $\lfloor w' \rfloor = \lfloor w \rfloor$ and both the order of fractional parts and integer bounds in t' are the same as in t , since

$$\lceil \text{var}_l(t') \rceil = \lceil \text{var}_l(s' + w') \rceil \leq \lceil \text{var}_l(s') \rceil + \lfloor w \rfloor + 1 + \delta = \lceil \text{var}_l(t) \rceil$$

by the second assumption on δ . The inequality in the other direction holds as well, and we get that $t' \sim t$ as required.

Case 3: $\{\text{var}_i(t)\} < \{w\}$ and there exists $j \notin R$ with $\{\text{var}_j(s')\} < \{\text{var}_i(s')\}$.

In this case let l be the variable with maximum fractional part in t , i.e. the last one which did not “jump” above an integer due to $\{w\}$. The variable with next bigger fractional part in s (and by \sim also in s') is $\text{var}_i(s)$, as depicted in Figure 8.

To transfer the move to s' , consider these two variables in s' as depicted in Figure 9 and let $\delta = \{\text{var}_i(s')\} - \{\text{var}_l(s')\}$.

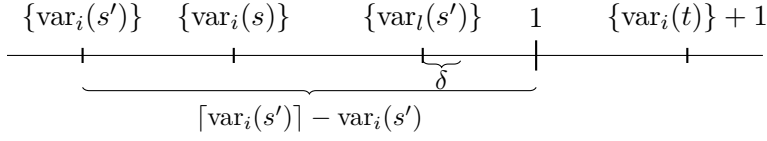


Figure 7: Lemma 5.2 Case 2

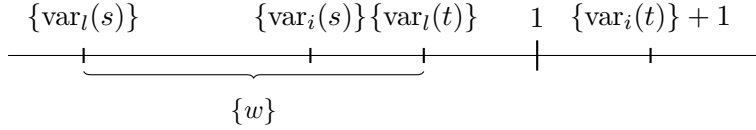


Figure 8: Lemma 5.2 Case 3 for s

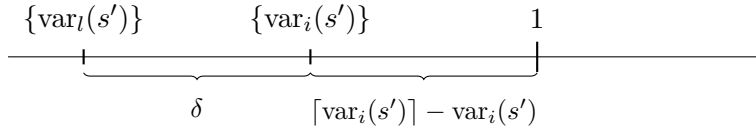


Figure 9: Lemma 5.2 Case 3 for s'

We set $w' = \lfloor w \rfloor + \lceil \text{var}_i(s') \rceil - \text{var}_i(s') + 0.9 \cdot \delta$. Again $\lfloor w' \rfloor = \lfloor w \rfloor$ and clearly i is the variable with smallest fractional part in t' by construction. As $s \sim s'$, the order of fractional parts in t and in t' is the same, and the integer bounds as well, thus $t \sim t'$. \square

5.1. Choosing Discrete Moves. Knowing that we can shift a single move and preserve \sim -equivalence, we proceed to 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.

However, for the purposes of constructing an inductive proof of existence of a good discrete strategy, it is not convenient to work, for a state s , simply with the maximal distance

$$\max_i \{ \min \{ \text{var}_i(s) - \lfloor \text{var}_i(s) \rfloor, \lceil \text{var}_i(s) \rceil - \text{var}_i(s) \} \}.$$

The reason is that for some moves it is impossible to keep this distance small for each variable and to go to an equivalent state as illustrated in Figure 10. In the depicted situation, if we move y_1 within ε -neighbourhood of \mathbb{Z} (below z and $z - 1$ depict integers), then y_0 leaves it. To give a more suitable notion of distance for a state, let us, 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 $> \frac{1}{2}$. as depicted in Figure 11.



Figure 10: Move where standard distance is necessarily increased.

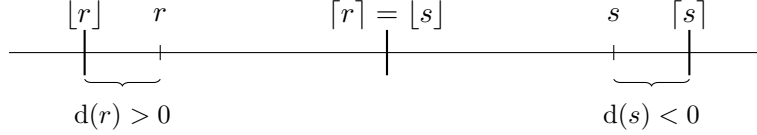


Figure 11: Notation for distances between real numbers and integers.

Please observe that for two real numbers $a, b \in \mathbb{R}_+$, it follows that

$$|d(a+b)| \leq |d(a)| + |d(b)|.$$

Also, we observe that

- if $|d(a) + d(b)| < \frac{1}{2}$, then $d(a+b) = d(a) + d(b)$;
- otherwise, if $d(a), d(b) = \frac{1}{2}$ or $d(a), d(b) = 0$, then $d(a+b) = 0$;
- otherwise, if $d(a), d(b) > 0$, then $d(a+b) = d(a) + d(b) - 1 < 0$;
- if $d(a), d(b) < 0$, then $d(a+b) = d(a) + d(b) + 1 > 0$.

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

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

This is illustrated in Figure 12, where k stands for an integer and y_0 to y_2 stand for the fractional parts of the values of the respective variables. In this example, y_0 has the smallest fractional part, i.e. the biggest one bigger than $\frac{1}{2}$ and y_2 has the biggest fractional part (less than $\frac{1}{2}$).

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 that the values of the variables only change within the same intervals.

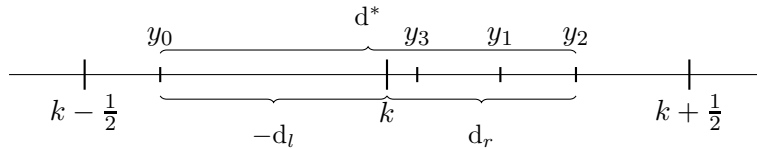


Figure 12: Maximal, minimal and total distances for a state.

Lemma 5.3. *Let s be a state with $d^*(s) \leq \frac{1}{4}$ and t be a successor of s , where (s, t) is allowed by $l = (I, \overline{C}, R)$. Then, for every $0 \leq \varepsilon < d^*(s)$, there exists a successor t'_+ of s such that*

- $t \sim t'_+$,
- (s, t'_+) is allowed by l , and
- $d^*(t'_+) \leq d^*(s) + \varepsilon$.

Proof. We assume that $d^*(t) > d^*(s) + \varepsilon$, otherwise we can take $t'_+ = t$. Let $w \in I$ be the increase in the (non-reset) values from s to t , i.e. $w = t -_R s$. We make a case distinction regarding the computation of $d^*(t)$.

Case 1: $d^*(t) = |d_l(t)|$.

We correct w in the following way: $w' = w + c - \varepsilon$, where $c = \min\{|d_r(t)|, |d(w)|\}$ if $d(w) < 0$ and $c = |d_r(t)|$ otherwise.

First, we have to show that $[w'] \in [w]$ and therefore $w' \in I$. Since $d_l(t) = d_i(t) = \text{var}_i(t)$ for one i , we can conclude from $|d(\text{var}_i(s) + w)| \leq |d(\text{var}_i(s))| + |d(w)|$ that $|d(w)| > \varepsilon$ and therefore $w' \geq w$, hence $w' \geq \lfloor w \rfloor$. Furthermore, $w' \leq \lceil w \rceil$. Otherwise, if $d(w) < 0$ then $w' = w + c - \varepsilon > \lceil w \rceil = w + |d(w)|$. This is a contradiction, since by definition $c \leq |d(w)|$.

If $d(w) > 0$, we also conclude $w' \leq \lceil w \rceil$, since $c - \varepsilon < \frac{1}{2}$.

Next, we have to show, that all variables that are not reset stay in the same interval. We consider the case, where all values of the variables are increased, therefore we know that $\text{var}_i(t'_+) \geq \lfloor \text{var}_i(t) \rfloor$ for all $i \notin R$. We now have to show that also $\text{var}_i(t'_+) \leq \lceil \text{var}_i(t) \rceil$. Let j be the index of the variable which is the closest to the integers (in this case), i.e. j , such that $d(\text{var}_j(t)) = d_r(t)$.

$$\begin{aligned} \text{var}_j(t'_+) &= \text{var}_j(s) + w' \\ &= \text{var}_j(s) + w + c - \varepsilon \\ &= \text{var}_j(t) + c - \varepsilon \\ &< \lceil \text{var}_j(t) \rceil = \text{var}_j(t) + |d_r(t)| \end{aligned}$$

Also, we have to show: $d^*(t'_+) \leq d^*(s) + \varepsilon$. We know that $|d_l(t)| - |d_r(t)| \leq d^*(s)$ and $d^*(t'_+) = |d_l(t'_+)| = |d(\text{var}_j(t'_+))|$ for one j and $\text{var}_j(t'_+) = \text{var}_j(s) + w + c - \varepsilon$. Hence, $d(\text{var}_j(t'_+)) = d_l(t) + c - \varepsilon$, since $|d_l(t) + c - \varepsilon| \leq \frac{1}{2}$. We can conclude that $d_l(t'_+) = d(\text{var}_j(t'_+)) \leq d^*(s) + \varepsilon$.

Case 2: $d^*(t) = |d_r(t)|$.

Subcase 1: $d(w) > 0$:

We correct w in the following way: $w' = w + (1 - c) - \varepsilon$, where $c = \max\{|d_l(t)|, |d(w)|\}$.

First, we have to show that $[w'] \in [w]$ and therefore $w' \in I$. Since $d_r(t) = d_i(t) = \text{var}_i(t)$ for one i , we can conclude from $|d(\text{var}_i(s) + w)| \leq |d(\text{var}_i(s))| + |d(w)|$ that $|d(w)| > \varepsilon$ and therefore $w' \geq w$, hence $w' \geq \lfloor w \rfloor$. Furthermore, $w' \leq \lceil w \rceil$. Otherwise, since $d(w) > 0$ and we assume that $w' = w + (1 - c) - \varepsilon > \lceil w \rceil = w + (1 - |d(w)|)$. This is a contradiction, since by definition $c \geq |d(w)|$.

Next, we have to show, that all variables that are not reset stay in the same interval. We consider the case, where all values of the variables are increased, therefore we know that $\text{var}_i(t'_+) \geq \lfloor \text{var}_i(t) \rfloor$ for all $i \notin R$. We now have to show that also $\text{var}_i(t'_+) \leq \lceil \text{var}_i(t) \rceil$. Let j be the index of the variable which is the closest to the integers (in this case), i.e. j , such

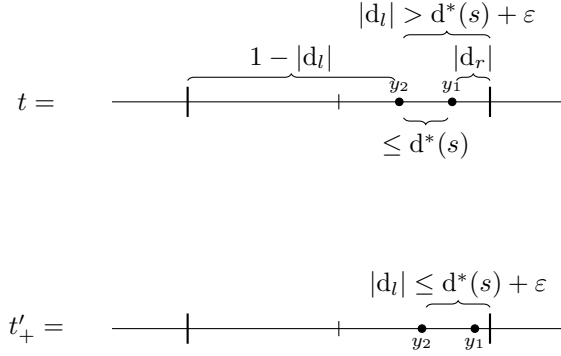


Figure 13: Case 1 illustration

that $d(\text{var}_j(t)) = d_l(t)$.

$$\begin{aligned}
\text{var}_j(t'_+) &= \text{var}_j(s) + w' \\
&= \text{var}_j(s) + w + (1 - c) - \varepsilon \\
&= \text{var}_j(t) + (1 - c) - \varepsilon \\
&< \lceil \text{var}_i(t) \rceil = \text{var}_j(t) + (1 - |d_l(t)|)
\end{aligned}$$

Also, we have to show: $d^*(t'_+) \leq d^*(s) + \varepsilon$. We know that $d_r(t) - d_l(t) \leq d^*(s)$ and $d^*(t'_+) = |d_r(t'_+)| = |d(\text{var}_j(t'_+))|$ for one j . $\text{var}_j(t'_+) = \text{var}_j(s) + w + (1 - c) - \varepsilon$. Hence, $d(\text{var}_j(t'_+)) = d_r(t) + (1 - c) + \varepsilon - 1 = d_r(t) - c + \varepsilon$. We can conclude that $d_r(t'_+) = d(\text{var}_j(t'_+)) \leq d^*(s) + \varepsilon$. by definition of c .

Subcase 2: $d(w) < 0$:

In this case, from $d^*(s) < \frac{1}{4}$ and $d^*(t) = d_r(t)$ it follows that $d(\text{var}_i(s)) < 0$ for all i . Thus, we set $w' = w + \lceil w \rceil - \varepsilon$ and the lemma holds.

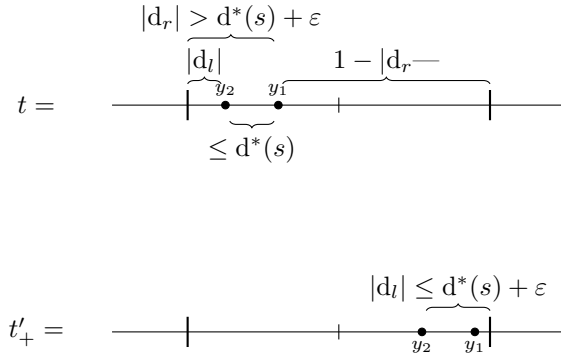


Figure 14: Case 2 illustration

Case 3: $d^(t) = d_r(t) + |d_l(t)|$.*

We correct w in the following way: $w' = w + c - \frac{\varepsilon}{2}$, where $c = \min\{|d_l(t)|, |d(w)|\}$.

First, we have to show that $\lceil w' \rceil \in [w]$ and therefore $w' \in I$. Since $d_r(t) = d_i(t) = \text{var}_i(t)$ for one i and $d_l(t) = d_j(t) = \text{var}_j(t)$ for one j , we can conclude from $|\text{d}(\text{var}_i(s) + w)| \leq |\text{d}(\text{var}_i(s))| + |\text{d}(w)|$ and $|\text{d}(\text{var}_j(s) + w)| \leq |\text{d}(\text{var}_j(s))| + |\text{d}(w)|$ and $|\text{d}(\text{var}_j(s) + w)| + |\text{d}(\text{var}_i(s) + w)| \leq |\text{d}(\text{var}_i(s))| + |\text{d}(w)| + |\text{d}(\text{var}_j(s))| + |\text{d}(w)| \leq d^*(s) + 2|\text{d}(w)|$ and $|\text{d}(\text{var}_j(s) + w)| + |\text{d}(\text{var}_i(s) + w)| > d^*(s) + \varepsilon$ therefore $|\text{d}(w)| > \frac{\varepsilon}{2}$. Hence, $w' \geq \lceil w \rceil$. Furthermore, $w' \leq \lceil w \rceil$, otherwise if $\text{d}(w) < 0$ then assume $w' = w + c - \frac{\varepsilon}{2} > \lceil w \rceil = w + |\text{d}(w)|$. Then $c - \frac{\varepsilon}{2} > |\text{d}(w)|$. Contradiction. Otherwise, if $\text{d}(w) > 0$, then $w' \leq \lceil w \rceil$, since by definition $c \leq \frac{1}{2}$.

Next, we have to show, that all variables that are not reset stay in the same interval. We consider the case, where all values of the variables are increased, therefore we know that $\text{var}_i(t'_+) \geq \lfloor \text{var}_i(t) \rfloor$ for all $i \notin R$. We now have to show that also $\text{var}_i(t'_+) \leq \lceil \text{var}_i(t) \rceil$. Let j be the index of the variable with $\text{d}(\text{var}_j(t)) = d_l(t)$.

$$\begin{aligned} \text{var}_j(t'_+) &= \text{var}_j(t) + w' \\ &= \text{var}_j(t) + w + c - \frac{\varepsilon}{2} \\ &= \text{var}_j(t) + c - \frac{\varepsilon}{2} \\ &< \lceil \text{var}_i(t) \rceil = \text{var}_j(t) + |\text{d}_l(t)| \end{aligned}$$

Thus we have to show: $d^*(t'_+) \leq d^*(s) + \varepsilon$. We know that $|\text{d}_r(t) - (1 + \text{d}_l(t))| \leq d^*(s)$ and $d^*(t'_+) = |\text{d}_l(t'_+)| = |\text{d}(\text{var}_i(t'_+))|$ for j such that $\text{d}(\text{var}_j(t)) = d_r(t)$. Also, $\text{var}_j(t'_+) = \text{var}_j(s) + w + c - \frac{\varepsilon}{2}$. We can conclude that $d^*(t'_+) \leq d^*(s) + \frac{\varepsilon}{2}$.

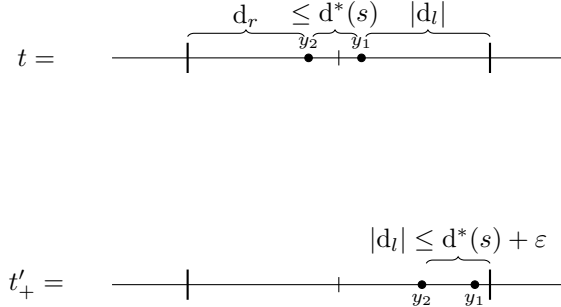


Figure 15: Case 3 illustration

□

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

Definition 5.4. 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 $d^*(s_{n+1}) \leq d^*(s_n) + \frac{\varepsilon}{2^{n+1}}$, and if for each i $s'_i \sim s_i$, then $\sigma(s_0 \dots s_n) \sim \sigma(s'_0 \dots s'_n)$.

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 5.5. To see that decreasing ε in each step is sometimes crucial, consider the game with one variable depicted in Figure 16. 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. $-y_0$, 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. Please note that the value of the game is 0.

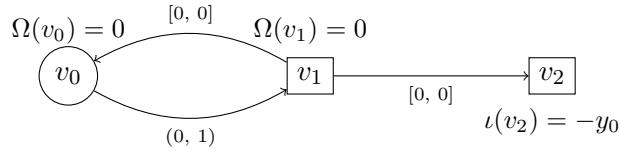


Figure 16: Game in which the values played must decrease.

We now extend the previous lemma to one that allows for the shifting of a whole move.

Lemma 5.6. *Let s be a state and t a successor of s , where (s, t) is allowed by l . Let s' be a state with $d^*(s') \leq \frac{1}{4}$, such that $s \sim s'$. Then, for every $\varepsilon > 0$, there exists a successor t' of s' allowed by l such that*

- $s' \sim t'$ and
- $d^*(t') \leq d^*(s') + \varepsilon$.

Proof. Since $s \sim s'$ and $t \in \text{succ}(s)$ is allowed by l , we know, by Lemma 5.2, that there exists a state $t' \in \text{succ}(s')$ allowed by the same label l , such that $t' \sim t$. We also know from Lemma 5.3 that, for every choice of ε , there exists $t_+ \in \text{succ}(s')$ such that $d^*(t_+) \leq d^*(s') + \varepsilon$ and $t' \sim t_+$. Since $t' \sim t$, this also means that $t_+ \sim t$, hence t_+ fulfils the requirements above. \square

We can conclude that discrete strategies allow for the approximation of game values.

Lemma 5.7. *Fix an ε -discrete strategy ρ_d of Player 1 – i in \mathcal{G} , $\varepsilon < \frac{1}{4}$. For every strategy σ of Player i there exists an ε -discrete strategy σ_d , such that, for every starting state s_0 with $d^*(s_0) < \frac{\varepsilon}{2}$, if $\pi(\sigma, \rho_d, s_0) = s_0 s_1 \dots$ and $\pi(\sigma_d, \rho_d, s_0) = s'_0 s'_1 \dots$, then $s_i \sim s'_i$ for all i .*

Proof. We only prove this lemma for Player 0, the case of Player 1 is analogous. We define σ_d inductively. Let s_0 be the starting state. If $\sigma(s_0) = s_1$, then by Lemma 5.6 there is a $s'_1 \sim s_1$ with $d^*(s'_1) \leq d^*(s_0) + \frac{\varepsilon}{4}$, and we set $\sigma_d(s_0) = s'_1$.

Let $h = s_0 \dots s_k$ and $h' = s'_0 \dots s'_k$ be finite play histories such that h is a prefix of $\pi(\sigma, \rho_d, s_0)$ and h' is consistent with ρ_d and σ_d as defined thus far. Note that $s_0 = s'_0$ and by inductive assumption $s_i \sim s'_i$ for $0 < i \leq k$, and $d^*(s_k) \leq \varepsilon(1 - \frac{1}{2^{k+1}})$. If $\sigma(s_0 \dots s_k) = s_{k+1} \in \text{succ}(s_k)$, then, by Lemma 5.6, there also exists a state $s'_{k+1} \in \text{succ}(s'_k)$ such that $s'_{k+1} \sim s_{k+1}$ and $d^*(s'_{k+1}) \leq d^*(s_k) + \frac{\varepsilon}{2}$. Thus, we set $\sigma_d(s'_0 \dots s'_k)$ to s'_{k+1} . For all other histories $h'' = s''_0 \dots s''_k$ with $s''_i \sim s_i$, we set $\sigma(h'') = s''_{k+1}$ for any s''_{k+1} equivalent with s_k , which exists by Lemma 5.2, and we can pick a discrete one if $d^*(s''_k) < \varepsilon$ by Lemma 5.6.

By construction, the strategy σ_d is discrete and if $\pi(\sigma, \rho_d, s_0) = s_0 s_1 \dots$ and $\pi(\sigma_d, \rho_d, s_0) = s'_0 s'_1 \dots$ then $s_i \sim s'_i$. \square

Proposition 5.8. *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*

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

Proof. Case 1: assume that

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

Then there exists a strategy $\sigma_d \in \Delta_0$ such that

$$\inf_{\rho \in \Delta_1} p(\pi(\sigma_d, \rho, s)) - \inf_{\rho \in \Gamma_1} p(\pi(\sigma_d, \rho, s)) > m.$$

Fix a strategy $\rho_{\text{inf}} \in \Gamma_1$, for which

$$p(\pi(\sigma_d, \rho_{\text{inf}}, s)) \leq \inf_{\rho \in \Gamma_1} p(\pi(\sigma_d, \rho, s)) + \varepsilon.$$

From Lemma 5.7, we know, that there is a discrete strategy $\rho_{\text{inf}_d} \in \Delta_1$ which is a discrete version of ρ_{inf} against σ_d . From the above, it follows that $p(\pi(\sigma_d, \rho_{\text{inf}_d}, s)) - p(\pi(\sigma_d, \rho_{\text{inf}}, s)) > m$. This is a contradiction, since we know from Lemma 5.7 that all states in both plays are equivalent, so for finite plays also the final states are equivalent, which means that the payoffs cannot differ by more than m as it is the highest occurring multiplicative factor in ι . If both plays are infinite, then, by definition of \sim , the payoffs are equal.

Case 2: assume that

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

By Theorem 3.5 every interval parity game is determined, thus

$$\sup_{\sigma \in \Gamma_0} \inf_{\rho \in \Gamma_1} p(\pi(\sigma, \rho, s)) = \inf_{\rho \in \Gamma_1} \sup_{\sigma \in \Gamma_0} p(\pi(\sigma, \rho, s)).$$

In the next section, we show that restricting to discrete strategies corresponds to playing a counter-reset game, and since these are again determined games, we get that

$$\sup_{\sigma \in \Delta_0} \inf_{\rho \in \Delta_1} p(\pi(\sigma, \rho, s)) = \inf_{\rho \in \Delta_1} \sup_{\sigma \in \Delta_0} p(\pi(\sigma, \rho, s)).$$

Therefore we can rewrite the assumption of this case as

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

Then there exists a strategy $\rho_d \in \Delta_1$ such that

$$\sup_{\sigma \in \Gamma_0} p(\pi(\sigma, \rho_d, s)) - \sup_{\sigma \in \Delta_0} p(\pi(\sigma, \rho_d, s)) > m.$$

Fix a strategy $\sigma_{\text{sup}} \in \Gamma_0$, for which

$$p(\pi(\sigma_{\text{sup}}, \rho_d, s)) \geq \sup_{\sigma \in \Gamma_0} p(\pi(\sigma, \rho_d, s)) - \varepsilon.$$

From Lemma 5.7, we know, that there again is a discrete strategy $\sigma_{\text{sup}_d} \in \Delta_0$ which is a discrete version of σ_{sup} against ρ_d . From the above, it follows that $p(\pi(\sigma_{\text{sup}_d}, \rho_d, s)) - p(\pi(\sigma_{\text{sup}}, \rho_d, s)) > m$, which again contradicts that all states in these two plays are equivalent. \square

6. COUNTER-RESET GAMES

In this section, we introduce counter-reset games and show, using the discretisation results from the previous section, that approximating the value of an IPG over \mathbb{Z}_∞ can be reduced to solving a counter-parity game. We then solve these games using an algorithm from [2].

By Proposition 5.8 above, we can restrict both players in a flat IPG 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). 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 or bigger than any of the (non-infinite) bounds in constraint intervals. This way, we can eliminate both ε 's and constraints and are left with the following games.

Definition 6.1. A *counter-reset* game is a flat interval parity game in which in each label $l = (I, \overline{C}, R)$ the constraints \overline{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.

Example 6.2. In Figure 6, we depict a simple counter-reset game. As usual, circles represent positions of Player 0 and boxes those of Player 1. Priorities, payoff functions, intervals and reset sets are also depicted as usual next to the corresponding nodes or above transitions. In this game, we have two variables, y_0, y_1 and as mentioned above, there are no constraints on these variables in counter-reset games, but they can be reset. The only choice in this game that Player 0 has is to increase all variables (“choose” 1 from $[1, 1]$) and Player 1 can do the same or end the game and get a payoff of $-y_0$. Since he wants to minimise, his best strategy is to loop as long as possible but not infinitely long, as the lowest priority on the according cycle is 0. Since he can achieve arbitrary small values this way, the value of this game (starting at v_0 or v_1) is $-\infty$.

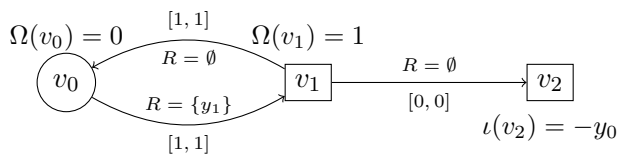


Figure 17: Simple counter-reset game

Lemma 6.3. 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:

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

Proof. Consider first the game $\mathcal{G}'' = n \cdot m \cdot \mathcal{G}$. By construction, the multiplicative factors in ι do not change and thus their maximal value in \mathcal{G}'' is still m . By Lemma 4.4, in all states

s holds

$$\text{val}\mathcal{G}(s) = \frac{\text{val}\mathcal{G}''(s)}{n \cdot m}.$$

Moreover, by Proposition 5.8 applied to \mathcal{G}''

$$\left| \text{val}\mathcal{G}''(s) - \sup_{\sigma \in \Delta_0} \inf_{\rho \in \Delta_1} p(\pi_{\mathcal{G}''}(\sigma, \rho, s)) \right| \leq m,$$

and therefore

$$\left| \text{val}\mathcal{G}(s) - \frac{\sup_{\sigma \in \Delta_0} \inf_{\rho \in \Delta_1} p(\pi_{\mathcal{G}''}(\sigma, \rho, s))}{n \cdot m} \right| \leq \frac{1}{n}.$$

We will now show how to construct the counter-reset game \mathcal{G}' with value equal to $\sup_{\sigma \in \Delta_0} \inf_{\rho \in \Delta_1} p(\pi_{\mathcal{G}''}(\sigma, \rho, s))$, i.e. to the value of \mathcal{G}'' when both players play ε -discrete strategies.

To this end, we first construct the game \mathcal{G}'_0 which still has constraints, but in which all intervals are $[k, k]$ for some $k \in \mathbb{N}$. The game \mathcal{G}'_0 is constructed from \mathcal{G}'' by replacing each position v by 3^M positions $v^{i_1 \dots i_M}$. The sequence $i_1 \dots i_M \in \{-1, 0, 1\}^M$ keeps track, for each variable, whether it is currently smaller, greater, or equal to an integer. The interval labels are now converted in the following way. If a move with interval $[n, n+k)$ and resets R is taken from a position $v^{i_1 \dots i_M}$ in \mathcal{G}'_0 and would lead to w in \mathcal{G}'' , then a sequence of moves with labels $[l, l]$ for each $n \leq l \leq n+k$ is added, with the l -labelled move leading to $w^{j_1 \dots j_M}$ such that:

- if one $j_k > i_k$ then all $j_k > i_k$ for $k \in \{0, \dots, M\}$, and the same if $j_k < i_k$ or $j_k = i_k$,
- if $l = n$ then each $j_k \geq i_k$ (interval was downwards-closed), and
- if $l = n+k$ then each $j_k < i_k$ (interval was upwards-open).

The situation for open, closed, and open-closed intervals is analogous. The plays which use discrete strategies in \mathcal{G}'' can now be directly transferred to plays in \mathcal{G}'_0 in which indeed in $v^{i_1 \dots i_M}$ the sign of the fractional part of y_j is equal to i_j . The same can be done in the other direction, as the constraints listed above allow to choose a value in the interval which leads to the appropriate change in the sign sequence. Therefore

$$\text{val}\mathcal{G}'_0 = \sup_{\sigma \in \Delta_0} \inf_{\rho \in \Delta_1} p(\pi_{\mathcal{G}''}(\sigma, \rho, s)).$$

To eliminate the constraints from move labels in \mathcal{G}'_0 we determine the highest non-infinite bound b which appears in these constraints (both on the left and on the right side of an interval). Then, we construct \mathcal{G}' as the synchronous product of \mathcal{G}'_0 with a memory of size $(b+2)^M$ which remembers, for each variable y_i , whether y_i is greater than b or equal to $b, b-1, \dots, 0$. With this memory, we resolve all constraints and remove them from move labels in \mathcal{G}' . \square

Counter reset games are another representation of a class of counter parity games, which were recently studied in [2], where an algorithm to solve such games was given, improving our previous decidability result [9].

Theorem 6.4 ([2]). *For any finite counter parity game \mathcal{G} and initial vertex v , the value $\text{val}\mathcal{G}(v)$ can be computed in 6EXPTIME. When the number of counters is fixed, the value can be computed in 4EXPTIME.*

Corollary 6.5. *For any finite counter reset game \mathcal{G} with a starting state s where all counters are integers, the value $\text{val}\mathcal{G}(s)$ can be computed in 6EXPTIME. With fixed number of counters, the value can be computed in 4EXPTIME.*

7. CONCLUSIONS AND FUTURE WORK

We conclude by completing the proof of our main Theorem 2.7. We first observe that, by Theorem 3.5, 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 4.2 and then into one over \mathbb{Z}_∞ by Corollary 4.5. By Lemma 6.3 the value of such a game can be approximated with arbitrary precision by counter-reset games, which we can solve by Corollary 6.5.

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. With the recent result on counter parity games, we are even able to provide an elementary algorithm – as the game \mathcal{G}'_n in Lemma 6.3 is doubly-exponential in \mathcal{G} and n , the combined complexity of the above procedure is 8EXPTIME (note the doubly-exponential increase compared to Corollary 6.5).

This complexity is very high and the complexity bound is not tight, thus we can formulate two immediate open problems: (1) can the exact value of $\llbracket\varphi\rrbracket^{\mathcal{K}}$ be computed? (2) what is the exact complexity of such a computation or its approximation? Another open question is whether we can use our methods for more general classes of games, e.g. for games with more complex payoff functions such as mean-payoff interval games. Furthermore, we are not only interested in theoretical complexity bounds but also in the practical applicability of quantitative model checking. This will require a more thorough algorithmic analysis of the problem. Also, since we reduce the problem to counter parity games, the implementation of a solver for this class of games is a necessary first step before we can exploit the methods presented in this paper in practice. However, even with further research needed to answer these challenges, our result lays the foundation for using quantitative temporal logics in the 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] Dietmar Berwanger, Lukasz Kaiser, and Simon Lessenich. Solving counter parity games. In *Proceedings of MFCS'12*, LNCS. Springer, 2012.
- [3] Patricia Bouyer, Thomas Brihaye, and Fabrice Chevalier. Weighted o-minimal hybrid systems are more decidable than weighted timed automata! In *Proceedings of LFCS'07*, volume 4514 of LNCS, pages 69–83. Springer, 2007.
- [4] Julian C. Bradfield and Colin Stirling. Modal logics and mu-calculi: an introduction. In *Handbook of Process Algebra*, pages 293–332. Elsevier, 2001.
- [5] Luca de Alfaro. Quantitative verification and control via the mu-calculus. In *CONCUR*, volume 2761 of LNCS, pages 102–126. Springer, 2003.
- [6] Luca de Alfaro, Marco Faella, and Mari elle Stoelinga. Linear and branching metrics for quantitative transition systems. In *ICALP*, volume 3142 of LNCS, pages 97–109. Springer, 2004.
- [7] Luca de Alfaro and Rupak Majumdar. Quantitative solution of omega-regular games. *J. Comput. Syst. Sci.*, 68(2):374–397, 2004.

- [8] Diana Fischer, Erich Grädel, and Lukasz Kaiser. Model checking games for the quantitative μ -calculus. *Theory Comput. Syst.*, 47(3):696–719, 2010.
- [9] Diana Fischer and Lukasz Kaiser. Model checking the quantitative mu-calculus on linear hybrid systems. In *Proceedings of the 38th International Colloquium on Automata, Languages and Programming, ICALP 2011 (2)*, volume 6756 of *Lecture Notes in Computer Science*, pages 404–415. Springer, 2011.
- [10] Thomas Gawlitza and Helmut Seidl. Computing game values for crash games. In *ATVA*, volume 4762 of *LNCS*, pages 177–191. Springer, 2007.
- [11] Hugo Gimbert and Wieslaw Zielonka. Perfect information stochastic priority games. In *Proceedings of ICALP'07*, volume 4596 of *LNCS*, pages 850–861. Springer, 2007.
- [12] Erich Grädel. Finite model theory and descriptive complexity. In *Finite Model Theory and Its Applications*, pages 125–230. Springer, 2007.
- [13] Thomas A. Henzinger, Pei-Hsin Ho, and Howard Wong-toi. Algorithmic analysis of nonlinear hybrid systems. *IEEE Transactions on Automatic Control*, 43:225–238, 1996.
- [14] 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.
- [15] 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.
- [16] Gerardo Lafferriere, George J. Pappas, and Shankar Sastry. O-minimal hybrid systems. *Mathematics of Control, Signals, and Systems*, 13:1–21, 2000.
- [17] Annabelle McIver and Carroll Morgan. Results on the quantitative μ -calculus $\text{qM}\mu$. *ACM Trans. Comput. Log.*, 8(1), 2007.
- [18] Vladimeros Vladimerou, Pavithra Prabhakar, Mahesh Viswanathan, and Geir E. Dullerud. Stormed hybrid systems. In *Proceedings of ICALP'08 (2)*, volume 5126 of *LNCS*, pages 136–147. Springer, 2008.