The Quantitative μ-Calculus

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH Aachen University zur Erlangung des akademischen Grades einer Doktorin der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Diplom-Informatikerin

Diana Fischer

aus Frechen

Berichter: Univ.-Prof. Dr. Erich Grädel
Prof. James Worrell
Univ.-Prof. Dr. Dr.hc. Wolfgang Thomas

Tag der mündlichen Prüfung: 23. April 2013

Abstract

This thesis studies a generalisation of the modal μ-calculus, a modal fixed-point logic that is an important specification language in formal verification. We define a quantitative generalisation of this logic, meaning that formulae do not just evaluate to true or false anymore but instead to arbitrary real values. First, this logic is evaluated on a quantitative extension of transition systems equipped with quantitative predicates that assign real values to the nodes of the system. Having fixed a quantitative semantics, we investigate which of the classical theorems established for the modal μ-calculus can be lifted to this quantitative setting.

The modal μ-calculus is connected to bisimulation, a notion of behavioural equivalence for transition systems. We define a quantitative notion of bisimulation as a distance between systems. First, we show that for systems that have a fixed maximal distance, the evaluation of formulae also differs by at most this distance, thus providing a quantitative version of the classical result that the modal μ-calculus is invariant under bisimulation. The converse direction does not hold for $Q\mu$ on arbitrary systems. However, as in the classical case, on finitely-branching systems the converse can be shown for the modal fragment and thus already quantitative modal logic characterises quantitative bisimulation on finitely-branching systems.

Next, we consider the model-checking problem which, given a system and a formula, is to decide whether the system is a model of the formula. In the quantitative world, this translates to computing the numerical value of a formula at a given node of the system. The model-checking problem for the modal μ-calculus can be solved by parity games, a class of infinite zero-sum graph games. We introduce a quantitative extension of these games and show that they are the corresponding model-checking games for our logic.

After establishing bisimulation invariance and developing model-checking games, we move to systems that are closer to the scenarios arising in practical applications. First, we consider discounted systems, i.e. systems where additionally the edges are labelled with quantities. It is not straightforward to define a negation operator in this setting that allows for the duality properties

needed for a game-based approach to model checking. We show that in this setting there is only one reasonable way to define it. Again, we define an appropriate extension of parity games and show that they correctly describe the evaluation of a discounted quantitative μ-calculus formula. Finally, we provide an algorithm for solving these games, thus also for model checking the quantitative μ-calculus on discounted systems.

In the final chapter, we go even further towards practical applications and evaluate the quantitative μ-calculus on a simple class of hybrid systems, namely initialised linear hybrid systems. We show how to approximate the value of a quantitative μ-calculus formula with arbitrary precision on such systems. We define a corresponding version of parity games and use the previously obtained results to prove that they are the correct model-checking games. Then we show how to simplify these games in several steps. We provide a detailed mathematical analysis of these games. In particular we introduce a new class of almost discrete strategies that permit us to simplify the games and to compute their values.

Zusammenfassung

Diese Dissertation behandelt eine Verallgemeinerung des modalen μ-Kalküls, einer wichtigen Spezifikationssprache in der formalen Verifikation. Die quantitative Erweiterung ist dergestalt, dass Formeln nicht länger nur zu *wahr* oder *falsch* ausgewertet werden können, sondern zu beliebigen reellen Zahlen. Wir werten diese Logik zunächst auf einer quantitativen Erweiterung von Transitionssystemen aus, in der jedem Knoten durch quantitative Prädikate reelle Werte zugewiesen werden. Anschließend untersuchen wir, welche klassischen Resultate für den μ-Kalkül sich sinnvoll in diesen quantitativen Rahmen übertragen lassen.

Eines dieser Resultate ist die enge Verbindung zwischen dem μ-Kalkül und der Bisimulation, einer Art von Verhaltensäquivalenz für Transitionssysteme. Wir führen quantitative Bisimulation als einen Abstand zwischen Transitionssystemen ein und zeigen, dass für Systeme, die einen festen Abstand haben, auch der Unterschied in der Auswertung von Formeln durch diesen Abstand begrenzt wird. Dies ist eine quantitative Version des klassischen Resultats, dass der modale μ-Kalkül invariant unter Bisimulation ist. Die Rückrichtung dieses Satzes gilt nicht für beliebige Systeme, aber wie im klassischen Fall zeigen wir, dass für endlich verzweigte Systeme diese Richtung bereits für die quantitative Modallogik gilt. Somit charakterisiert quantitative Modallogik quantitative Bisimulation für endlich verzweigte Systeme.

Weiterhin betrachten wir das Model-Checking-Problem, d.h. die Frage ob für ein gegebenes System und eine Formel gilt, dass das System Modell der Formel ist. Für den quantitativen Fall lässt sich dies übersetzen in die Berechnung der Auswertungsfunktion einer Formel für ein gegebenes System. Im klassischen Fall kann dies durch die Übersetzung in Paritätsspiele gelöst werden, einer Klasse von unendlichen Graphspielen. In dieser Arbeit führen wir eine quantitative Erweiterung dieser Spiele ein und zeigen, dass diese die geeigneten Model-Checking-Spiele für den quantitativen μ-Kalkül sind.

Schließlich beschäftigen wir uns mit Anwendungsszenarien für den quantitativen μ-Kalkül, wobei wir zuerst eine entsprechend erweiterte Logik auf *discounted systems* betrachten. Dies sind quantitative Systeme in denen auch

die Kanten mit reellen Werten beschriftet sind. Die Definition eines geeigneten Negationsoperators – welcher entscheidend für den spielbasierten Zugang zum Model-Checking-Problem ist – ist in diesem Fall nicht offensichtlich. Wir zeigen, dass es nur eine sinnvolle Art gibt diesen zu definieren. Wie bereits zuvor führen wir eine passende Erweiterung von Paritätsspielen ein und zeigen, dass sie die geeigneten Model-Checking-Spiele für diesen Fall sind. Anschließend zeigen wir, wie sich diese Spiele algorithmisch lösen lassen.

Im letzten Kapitel werten wir den quantitativen μ-Kalkül auf einer einfachen Klasse von hybriden Systemen aus. Wir zeigen, dass der Wert einer quantitativen μ-Kalkül-Formel mit beliebiger Präzision auf diesen Systemen berechnet werden kann. Dazu nutzen wir wieder die Beschreibung durch entsprechende Paritätsspiele aus, sowie die Resultate aus den vorherigen Kapiteln. Die Berechnung läuft in mehreren Reduktionsschritten ab. Wir führen eine detaillierte mathematische Analyse dieser Spiele durch. Insbesondere definieren wir eine neue Klasse von Strategien, die uns erlaubt, die Spiele zu vereinfachen und ihre Werte zu berechnen.

Acknowledgements

Working on the topics which comprise this thesis has been an exciting adventure and I would like to take this opportunity to thank all the people that supported me along the way. This thesis would not exist without them.

First, I would like to express my deep gratitude to my supervisor Erich Grädel for his excellent lectures and seminars which got me interested in this line of research in the first place, for giving me the freedom to pursue the topics that I was interested in, for his insights, encouragement and giving me support whenever I needed it. I am also deeply grateful to my collaborator Łukasz Kaiser. I gained so much from our joint work over the years, doing research with him was an inspiring, educational and also very enjoyable experience. I thank Wolfgang Thomas who was my co-supervisor in the DFG research training group AlgoSyn for his encouragement and support, and for motivating us through his positive attitude and enthusiasm towards research and all other facets of academic life. I am grateful to Ben (James) Worrell for agreeing to be an external reviewer and for giving me very helpful remarks on the first draft of this thesis. Furthermore, I would like to thank Berthold Vöcking and Stefan Kowalewski for taking the time to be part of my exam committee. I had the wonderful opportunity to visit Luke Ong and his research group in Oxford during my doctoral studies and I am thankful for their hospitality and the great time that I had there.

Tobias Ganzow and I shared an office for several years and I enjoyed our discussions which covered research, academia, but also the more mundane aspects of life. I am thankful to Faried Abu Zaid, Simon Leßenich, Stefan Repke, Wied Pakusa and Roman Rabinovich for proofreading a draft of this thesis. I would also like to thank the other members of the MGI-I7 family, Alex, Bernd, Christof, Daniel, Dietmar, Frank, Ingo, Jan, Jörg, Kari, Marcus, Martin L., Martin Z., Michael H., Michael U., Michaela, Namit, Vince and Wladi. Life at the department with them was always enjoyable and I will miss the coffee breaks in which we discussed not only our research and teaching activities, but also all kinds of other topics, solved logic puzzles and learned to play Go together. I am grateful to Marianne Kuckartz, Silke Cormann, Helen Bolke-

Hermanns and Rolf Eschmann for taking care of the administrative sides of academia and thus making our lives so much easier.

I would also like to thank Christoph for his support, encouragement, and especially for his understanding in the final days of writing up my thesis. I am deeply grateful to my parents, Heidi and Rudi, for their love and unconditional support and for knowing that I can always rely on them. Last but certainly not least, I would like to express my gratitude to my brother André for always being there for me whenever I need him and I dedicate this thesis to him.

Contents

List of Figures

1 Introduction

Formal verification is one of the most important challenges in theoretical computer science. One of its central tasks is to provide formal methods to assure that a program or a system behaves according to a given specification. Failure to predict or guarantee the behaviour of a program can lead to anything from minor annoyance to major safety problems. For example, it is indispensable that the software running in a plane that assists the pilot with the landing procedure runs smoothly and does not exhibit faulty behaviour. The same can be said, for example, for the software that controls a nuclear power plant or other safety-critical systems. If these programs are only verified by running test cases or scenarios – however exhaustive – one can never be completely sure that all possible behaviours have been covered and that the software will never produce an error. This is where formal verification comes into play. In order to use formal methods we first have to model systems in an abstract way. Furthermore, we have to provide a language in which we can specify the desired behaviour. The languages of choice in verification are usually temporal logics and one of the most prominent verification techniques is model checking, i.e. proving that a model of the system fulfils a logical formula which represents its specification. Two prominent examples of temporal logics are the linear-time temporal logic LTL and the branching-time temporal logic CTL (computation tree logic). These logics allow for specifying properties of runs of programs. For example, the liveness property "a deadlock should never occur" could be specified in a temporal logic. In linear time logics a run is a linear succession of events that can be viewed as a time line and thus statements are implicitly universally quantified. In branching-time logics we view the execution of a program as a tree, for every possible event, we branch out. This tree describes all possible future behaviours and every run through this tree can be viewed as a linear time line.

The modal μ-calculus, introduced by Kozen in [31], is a modal logic with least and greatest fixed-point operators and is expressive enough to subsume LTL and CTL and even CTL*, a generalisation of both, see e.g. [6]. This makes the modal μ-calculus one of the most powerful logical formalisms in formal verification. In addition to its great expressive power, the modal μ-calculus has nice model-theoretic properties.

1.1 The Modal μ-Calculus

The modal μ-calculus, L_μ, is a modal logic equipped with least and greatest fixed-point operators. In modal logic, we build formulae from a set of atomic propositions, a negation operator $\neg$ (not), logical operators $\wedge$ (and) and $\vee$ (or), and modal operators $\Box$ (box) and $\Diamond$ (diamond). A formula is evaluated over a transition system, i.e. a directed graph where the nodes are labelled with atomic propositions. We can describe features of transition systems by modal formulae. Atomic formulae are just propositions that are either true or false at a node and we have the intuitive interpretation of $\neg$, $\wedge$ and $\vee$. The modal operators allow to speak about the successors of a node, where $\Box\varphi$ requires all successors to fulfil the formula φ and $\Diamond\varphi$ requires the existence of a successor where φ holds. An example formula in modal logic is $\varphi = \Diamond(P \wedge \neg Q)$ which holds at a node if there exists a successor where the atomic property P holds and Q does not. Modal logic can be seen as a fragment of first-order logic and thus lacks a recursion mechanism which is needed e.g. to express reachability properties.

The modal μ-calculus introduces recursion by adding least and greatest fixed points of definable operators. These fixed-point operators are denoted in the syntax by the Greek letters μ (least fixed point) and ν (greatest fixed point). We do not formally describe the semantics of L_μ here but instead give some example formulae to illustrate what is expressible in this logic. For example, reachability is now expressible: "eventually P" is expressed in L_μ as $\mu X.(P \vee \Diamond X)$. Also, the requirement that a system be non-terminating, i.e. that the system will never reach a point from which there are no outgoing edges is expressible as $\nu X.(\Diamond X)$. Another example of a temporal property is "Q until P", or in L_μ, $\mu X.(P \vee (Q \wedge \Diamond X))$. This holds if either P holds immediately on the initial node, or it holds at a later node and on all nodes on paths in-between the property Q is true.

The close connection to games is a fundamental aspect of logics. The evaluation of logical formulae can be described by model-checking games, played by two players on an arena which is formed as the product of a structure $\mathcal{K}$ and a formula ψ. One player (Verifier) attempts to prove that ψ is satisfied in $\mathcal{K}$ while the other one (Falsifier) tries to refute this. For the modal μ-calculus L_μ, model checking is described by *parity games*, and this connection is of crucial importance for the algorithmic evaluation and the applications of the μ-calculus. Indeed, most competitive model checking algorithms for L_μ are based on algorithms to solve the strategy problem in parity games [29]. Furthermore, parity games enjoy nice properties like positional determinacy and can be intuitively understood: often, the best way give an intuitive meaning to a μ-calculus formula is to look at the associated game. In the other direction winning regions of parity games (for a fixed number of priorities) are definable in the modal μ-calculus. A summary of the results about L_μ, parity games and model-checking can be found in [24].

The modal μ-calculus is linked to a classical relation called *bisimulation*. Bisimulation has been introduced as a notion of behavioural equivalence between systems. It has its origin in concurrency theory where it was used to formalise the similarity of the behaviour of processes [25, 36]. Hennessy and Milner additionally proposed a logical characterisation of bisimulation. Independently, bisimulation was studied in the context of classical logics and Kripke structures, and was used to identify the class of formulae of first-order logic which are equivalent to formulae in modal logic [38, 39]. There are different ways to define bisimulation: originally it was defined as a fixed point of a monotone self-map on a complete lattice. Van Benthem gave a relational definition, and it can also be defined in terms of a two-player game. As stated above, bisimulation captures behavioural equivalence between transition systems. Intuitively speaking, two systems are bisimilar if we can see the same behaviour on all runs through the systems, even though the general shape of the systems might be different. The modal μ-calculus is invariant under bisimulation. This means that a specification written in L_μ cannot distinguish between systems that are bisimilar. This is a desirable property for a specification language as one often is not interested in e.g. the specific implementation of a program but mainly in its behaviour. The converse direction of this statement, i.e. indistinguishability in L_μ implies bisimilarity, does not hold for the modal μ-calculus on general systems [5]. It is true, however, for infinitary modal logic, an extension of modal logic that admits infinite conjunctions and disjunctions and subsumes the modal μ-calculus (on classes of systems of bounded cardinality). For finitely-branching systems, already modal logic *characterises* bisimulation, i.e. finitely-branching systems that cannot be distinguished by a formula in modal logic are bisimilar [23].

Bisimulation can not only be defined as a relation but equivalently as a game between two players, Spoiler and Duplicator. In the game, a pebble is placed on each system. In each round Spoiler challenges Duplicator by moving one of the pebbles to show a difference between the systems, and Duplicator has to match his move and prove that the systems are still bisimilar. The game-based description enables us to use tools from game theory to prove results in logic, an approach that we already used for the model checking problem. For a formal treatment of modal logics, L_μ, and the classical results on bisimulation we refer the reader to [23, 5].

1.2 Quantitative Logics

There is an obvious motivation to extend classical two-valued formalisms to quantitative ones: systems in the real world have quantities in them. If we model a real system, for example an engine, there are properties that can naturally be described using quantities, such as the amount of fuel, temperature, pressure and so on. When querying the system, it seems natural to ask "what is the highest temperature on each run of the system?" or "how much fuel is left after n steps?". The answer to such queries is a number and it is desirable not having to rephrase them as yes-or-no questions. While there is a great motivation to define quantitative logics, one has to be careful in doing so. In many areas of practice where logical formalisms are applied, one can observe that the definitions of quantitative formalisms are often ad-hoc and not well thought-out. Quantitative formalisms often lack the clean mathematical theory their qualitative counterparts enjoy and thus also lose their nice algorithmic properties. Furthermore, these mathematical properties are needed for a formal treatment, and thus the wrong quantitative extension can make it impossible to use classical approaches to formal verification.

The word *quantitative* can have different meanings when applied to logical formalisms. It can refer to a specific quantity, such as time or probability, or it can just be an abstract number. There are application areas for all of these approaches and so, unsurprisingly, there are already many different proposals for quantitative versions of logics, especially of temporal logics and the μ-calculus. In the broadest sense *quantitative* means that instead of a logic being two-valued, i.e. allowing only for the evaluations true or false, formulae can evaluate to arbitrary quantities.

In many of the proposed extensions of temporal logics, the term quantitative is used as a synonym for probabilistic. In these works, a logic is interpreted over probabilistic transition systems [33], or used to describe winning conditions in stochastic games [12, 7, 22]. Other variants introduce quantities by allowing discounting in the respective version of a "next"-operator for qualitative transition systems [7], Markov decision processes and Markov chains [8], and for stochastic games [11]. In [34], the μ-calculus is also interpreted over probabilistic transition systems, and it is shown that the value of a formula can be described by an appropriate two-player stochastic game. The proof techniques in [34] are adapted from the unfolding technique that we also use in this thesis.

We already stated that the modal μ-calculus is linked to a notion of behavioural equivalence called bisimulation. In the quantitative setting, bisimulation has already been interpreted in the context of behavioural pseudometrics. The pseudometric allows for defining a distance between states representing the similarity of their behaviours. In [40] quantitative bisimulation equivalence has been studied for (edge-)labelled transition systems and been given three characterisations: a fixed-point, a logical and a co-algebraic one. Their logic allows for labelled modalities and the distance between edge label and modal label describes the requirement on the following sub formula. In [10], the authors study metric transition systems (where propositions can take values from arbitrary metric spaces) and quantitative versions of simulation and bisimulation. They also give a definition of a pseudometric using a fixed-point characterisation and in terms of a positive quantitative modal logic. In [13], another fixed-point characterisation of a bisimulation pseudometric can be found.

1.3 Organisation and Main Results

In this thesis *quantitative* is an abstract term and has no pre-defined semantics. To be able to avoid overly abstract notions however, we chose to interpret quantitative as real-valued, so in our formalism formulae can evaluate to real values. We took our inspiration from a work by de Alfaro, Faella and Stoelinga [9, 10]. Their quantitative μ-calculus is an extension of L_μ (without negation and thus duality of operators). The semantics of this logic is given by metric transition systems, i.e. transition systems where nodes are labelled with quantities from arbitrary metric spaces. Additionally, they allow discounting in the modal operators, i.e. multiplying by a positive factor less than 1. They study this logic and a quantitative version of LTL in connection with a trace and bisimulation distance, a quantitative generalisation of trace equivalence and bisimulation, and they show a characterisation result for its modal fragment on finitely-branching systems.

We are interested in developing a quantitative model theory of the μ-calculus. As mentioned above, we restrict ourselves to real values. Also, we are not satisfied with a logic without negation as duality of operators is crucial for a game-based approach to model checking and negation is a part of a full logic.

Having settled on the reals, there are still many design choices to be made. Transition systems can be made quantitative to different degrees. We can label only the nodes with real values or we can allow labels on both nodes and edges. Also the logic can be made quantitative to different levels. Either we just allow quantitative predicates or we also allow a version of discounting in the formulae, i.e. multiplying or adding a real value. As we discussed above though, we have to be careful how we define the quantitative version of the μ-calculus so that we retain the classical properties that we expect from it – at least to a certain extent. To illustrate the consequences of the different choices in the quantitative design, we will treat the above mentioned cases separately.

Having defined a quantitative extension of L_μ which we call the quantitative μ-calculus $Q\mu$, we can now ask which of the classical theorems can be lifted into this quantitative setting and how the different scenarios compare to each other. Obviously, we cannot use the classical notions anymore, so we have define quantitative extensions of bisimulation and parity games. We also formulate quantitative versions of the classical theorems tailored to this setting. After establishing some of the classical results for the quantitative μ-calculus $Q\mu$, we discuss algorithms to evaluate it on more complex systems. We start by investigating discounted systems, and then go on to provide a model-checking algorithm for the quantitative μ-calculus on a class of hybrid systems.

Structure of this Thesis

The general structure of this thesis is as follows. First, we define the quantitative μ-calculus on simple quantitative transition systems, i.e. graphs where only the nodes are labelled with real values, and investigate a quantitative version of bisimulation. Then, we concern ourselves with model-checking, and define a quantitative version of parity games and prove that they correctly describe the evaluation of a formula. Towards practical applications, we consider more complex systems and make the logic more powerful. We discuss model-checking for discounted systems, i.e. quantitative transition systems where also the edges are labelled. Finally, we extend the model-checking algorithm even to a class of hybrid systems.

In Chapter 2, we introduce the quantitative μ-calculus, the heart of this thesis. We illustrate the kind of properties this logic is able to express. Then, we explore a quantitative version of bisimulation and illuminate its relation to our logic. We show a result similar to the classical invariance theorem for L_μ and the characterisation theorem for modal logic on finitely-branching systems.

First, we define a simple quantitative extension of Kripke structures. These quantitative transition systems are directed graphs where the nodes are labelled with quantitative predicates, i.e. functions that assign real values to each node.

Next, we define our quantitative version of the modal μ-calculus which we call the quantitative μ-calculus $Q\mu$. The syntax of this logic is similar to the classical μ-calculus. The formulae of $Q\mu$ are evaluated on quantitative transition systems and yield a function from the nodes of a transition system to the reals extended with ∞ and $-\infty$. We give a few examples for the evaluation of modal operators and fixed-point formulae to familiarise the reader with our logic.

Then, we introduce a quantitative version of bisimulation similar to the bisimulation distance by de Alfaro, Faella and Stoelinga [10, 9]. Although the resulting notion is similar, our definition differs from theirs. They define bisimulation as a least fixed point of a quantitative operator. We define bisimulation as a relation and as a game, and show that the notions coincide as in the classical case. Then, we extend the classical results. We establish that $Q\mu$ is invariant under quantitative bisimulation which means that if two systems are r-bisimilar, the evaluations of formulae in $Q\mu$ also differ by at most r, for a $r \in \mathbb{R}$. As previously stated, the reverse direction fails for $Q\mu$ as in the classical case. For finitely-branching systems, however, the reverse is true already for quantitative modal logic QML, the fragment of $Q\mu$ without fixed-point operators. If the evaluation of QML formulae of two finitely-branching systems differ by at most r, then these systems are r-bisimilar. This means that QML characterises quantitative bisimulation for finitely-branching systems. The proofs in this chapter make heavy use of the game-based description of quantitative bisimulation.

In Chapter 3, we explore how the classical relationship between parity games and the μ-calculus can be lifted into the quantitative setting. To this end, we introduce a quantitative version of parity games and prove that they are the appropriate model-checking games for Qμ. Quantitative parity games extend classical parity games by adding payoff rules for finite plays while infinite plays are still decided by the occurring priorities. This means that now we have no winner or loser anymore, but a quantitative payoff for each play. This changes the objectives of the two players, Verifier now wants to maximise the payoff, while Falsifier wants to minimise it. In this chapter, it becomes obvious why we could not forgo negation and thus duality of operators in our definition of Qμ earlier. Without the appropriate closure and duality properties – closure under negation, De Morgan equalities, quantifier and fixed point dualities – we would not be able to use the game-based approach.

Quantitative parity games are the model-checking games for Qμ. However, they turn out to be more complicated than their qualitative counterparts. They do not admit optimal strategies, but on the positive side we can show that they allow for positional ε-optimal strategies, i.e. simple strategies that do not need any memory to give an outcome close to the optimal value. The fact that we cannot guarantee optimal strategies makes the proof that the model-checking games correctly describe the value of a formula considerably harder than in the classical case. As in the classical setting, model-checking games lead to a better understanding of the semantics and expressive power of the quantitative μ-calculus. In the last section of this chapter, we show how we can prove the bisimulation invariance theorem of the previous section entirely using games, i.e. model-checking games and bisimulation games.

In Chapter 4, we introduce an extended version of the quantitative μ-calculus and quantitative parity games. We extend the systems by allowing discounts, i.e. real values, also on the edges of the quantitative transition systems. Furthermore, we equip the logic with a discount operator, i.e. we allow multiplication by a positive real value. Equivalently, we could use addition of a real value as the two semantics are interchangeable, which we also explain in more detail.

This version of Qμ was the first one we introduced in [16] and later explored in more detail in [17]. Here, we decided to treat the discounted and non-discounted version separately, therefore simplifying our original proofs which dealt with determinacy and correctness of model-checking directly in the discounted setting. Here, we omit these direct proofs and show instead how the correctness of the model-checking theorem in the discounted case follows from the corresponding theorem in the non-discounted case. First, we have to treat negation. If we allow for discounting, the choice of negation and how to ensure the duality of operators is not as obvious as one might expect. We show that there is only one reasonable way to define negation in our context.

Having done so, we can then define an appropriate extension of quantitative parity games. Unsurprisingly, they allow for the labelling of edges in the game graphs. This makes the rules for calculating payoffs of finite plays slightly more complicated. Although these games are only a minor extension of quantitative parity games, they behave in a substantially different way. Obviously, they also do not enjoy optimal strategies, but now they do not allow ε-optimal positional strategies anymore. Even worse, they do not even admit bounded-memory ε-optimal strategies for a given ε. Nevertheless, we show that these are the right model-checking games and that they correctly describe the value of a discounted Qμ formula. Then, we make use of the fact that we proved the corresponding theorem for quantitative parity games (of any cardinality) by showing how discounted games can be seen as a compact representation of these games. In the discounted setting, we can go even further and show that the model-checking correspondence goes both ways: the value of a formula on a structure coincides with the value of the associated model-checking game, and conversely, the value of a quantitative parity game (with a fixed number of priorities) is definable in the quantitative μ-calculus. Unfortunately, the corresponding quantitative formula uses a trick that needs discounting and so this does not work for the non-discounted calculus of the previous chapters.

In Chapter 5, we apply our formalism to a restricted class of hybrid systems. Hybrid systems are a combination of discrete transition systems and dynamically changing continuous variables. While they are an obvious candidate for practical applications, they are extremely difficult to handle and almost all interesting questions about them are undecidable [27]. Although this is clearly an area where formal verification methods can be applied, there have not yet been many attempts for quantitative model checking of such systems. Quantitative verification is mostly done by simulation and thus lacks the guarantees that can be given by model-checking techniques.

As mentioned before, it would be impossible to attempt formal verification on the class of all hybrid systems as even simple properties such as reachability are undecidable. For this reason, we consider the restricted class of initialised linear hybrid systems, which is one of the largest classes of hybrid systems with decidable temporal logic. Linear in this context means that the general differential equations used to describe the changes of continuous variables are linearised. This approximation alone does not lead to decidability though and so we also need the systems to be initialised. Initialised systems have to reset a variable if the evolution rate of that variable changes during a discrete transition.

In this chapter, we show how to approximate the value of hybrid Qμ on initialised linear hybrid systems. These results have been published in [18, 19]. First, we define linear hybrid systems in our setting. Then, we give the syntax of hybrid Qμ, a slight extension of Qμ which allows to use continuous variables in the same way as quantitative predicates. Afterwards, we describe a corresponding version of quantitative parity games which we call interval parity games and which can be used to model-check hybrid Qμ. As in the discounted case, these games can be seen as a compact representation of infinite parity games, a fact that we explain in detail.

We continue our investigation by showing how to simplify these games to ones where the linear coefficients are all 1. These games look similar to timed games but unfortunately behave quite differently, as we point out. Thus, we cannot use the classical region-graph construction. Instead we use a new class of almost discrete strategies to simplify these games even further. The resulting class of games, called counter-reset games, is a special case of counter parity games that were recently solved by Berwanger, Kaiser and Leßenich, [30]. Altogether, we show how we can compute the value of a Qμ formula with arbitrary precision on initialised linear hybrid systems. As the μ-calculus subsumes LTL, this result properly generalises a previous result on model checking LTL on such systems [26, 27].

1.4 Impact

The quantitative μ-calculus and the results on quantitative parity games that we present in this thesis have already been applied and extended, confirming the usefulness of this approach.

In [30], a variation of Qμ has been introduced that replaces quantitative predicates with terms from counting monadic-second order logic which, e.g., allows to express boundedness questions. It is evaluated on structured transition systems, i.e. systems where nodes are labelled with relational structures, and thus generalise many systems that occur in theoretical computer science such as pushdown automata, Turing machines, term and graph rewriting systems. To model-check this logic, Kaiser and Leßenich also use quantitative parity games and our result from Chapter 3. By additionally generalising a decomposition result, they show decidability of counting $Q\mu$ on tree-producing pushdown systems, a generalisation of pushdown systems and regular tree grammars. They re-prove and extend previous results [3, 21] on model-checking with unboundedness conditions on pushdown systems. In this case, the use of $Q\mu$ and model-checking games allows to obtain stronger results in a more systematic way and simplifies the proofs.

2 The Quantitative μ-Calculus Qμ

In this chapter we present a quantitative generalisation of the classical modal μ-calculus which was introduced by Kozen in [31]. We extend the classical two-valued formalism to one with infinitely many real values. A formula of the quantitative μ-calculus does not evaluate to just true or false anymore but to an arbitrary real number.

The classical μ-calculus is evaluated over transition systems, i.e. labelled directed graphs. First, we introduce a simple quantitative version of these structures. Quantitative transition systems are directed graphs where the nodes are labelled with quantitative, i.e. real-valued predicates. Then, we give the syntax and semantics for the quantitative μ-calculus, Qμ, and briefly discuss its fragment quantitative modal logic.

We proceed by introducing a quantitative version of bisimulation, the classical notion which describes behavioural equivalence between transition systems. We define quantitative bisimulation as a distance between transition systems in two ways, as a quantitative relation and as a quantitative two-player game and show that the notions are equivalent. Furthermore, we prove that – as in the classical world – the quantitative μ-calculus is invariant under this relation. Invariance in the quantitative setting means that a certain fixed bisimulation distance between two transition systems guarantees that also the evaluations of formula on these systems only differ by at most this distance. As in the classical case, the reverse direction does not hold for the μ-calculus [5], but if we restrict the models to systems that are finitely-branching, we can show a characterisation result already for quantitative modal logic. In the classical world, if finitely-branching transition systems are indistinguishable for modal logic, then they are bisimilar. This translates in the quantitative setting to the following. If the evaluation of all formulae of quantitative modal logic on two transition systems only differ by a certain number, then they have at most this bisimulation distance.

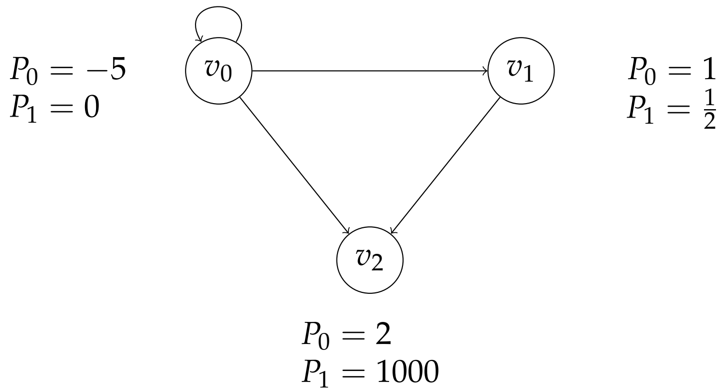


Figure 2.1: A simple quantitative transition system $\mathcal{Q} = (V, E, P_0, P_1)$

2.1 Syntax and Semantics

Let us fix some notation first. In the sequel, $\mathbb{R}_0^+$ is the set of positive reals including 0, $\mathbb{R}_\infty$ is the set of real numbers extended with a minimal and a maximal element, i.e. $\mathbb{R}_\infty := \mathbb{R} \cup \{-\infty, \infty\}$ and $\mathbb{R}_\infty^+$ is $\mathbb{R}_0^+ \cup \{\infty\}$. If not defined otherwise, we denote by $I \subset \mathbb{N}$ a finite index set.

Definition 2.1. A *quantitative transition system (QTS)* is a tuple

$$\mathcal{Q} = (V, E, \{P_i\}_{i \in I}),$$

consisting of a directed graph (V, E), where V is the set of nodes, $E \subseteq V \times V$ is the set of (directed) edges and $P_i : V \to \mathbb{R}_\infty$ for $i \in I$ are predicate functions that assign real values to each node.

A transition system is *qualitative* if all functions P_i assign only the values $-\infty$ or ∞, i.e. $P_i : V \to \{-\infty, \infty\}$, where $-\infty$ stands for false and ∞ for true. We call a transition system $\mathcal{Q}$ *pointed*, if it has a designated initial node v and denote this by $(\mathcal{Q}, v)$.

Example 2.2. In Figure 2.1, we depict a simple quantitative transition system $\mathcal{Q} = (V, E, P_0, P_1)$ with two quantitative predicates P_0 and P_1.

We now introduce a quantitative version of the modal μ-calculus to describe properties of quantitative transition systems.

Definition 2.3. Given a set $\mathcal{X}$ of fixed-point variables X, predicate symbols $\{P_i\}_{i\in I}$, and constant symbols $c \in \mathbb{R}$, the formulae of the *quantitative μ-calculus* (Qμ) are built in the following way.

(1) $P_i + c$ is a Qμ-formula,

(2) X is a Qμ-formula,

(3) if φ is a Qμ-formula, then so is $\neg\varphi$,

(4) if φ, ψ are Qμ-formulae, then so are $(\varphi \wedge \psi)$ and $(\varphi \vee \psi)$,

(5) if φ is a Qμ-formula, then so are $\Box\varphi$ and $\Diamond\varphi$,

(6) if φ is a formula of Qμ, then $\mu X.\varphi$ and $\nu X.\varphi$ are formulae of Qμ given that X occurs only positively (i.e. under an even number of negations) in φ.

Formulae of Qμ are interpreted over quantitative transition systems. For a transition system $\mathcal{Q}$, let $\mathcal{F}$ be the set of functions $f : V \to \mathbb{R}_\infty$. We say for $f_1, f_2 \in \mathcal{F}$ that $f_1 \leq f_2$ if $f_1(v) \leq f_2(v)$ for all $v \in V$. $(\mathcal{F}, \leq)$ forms a complete lattice with the constant functions $f = \infty$ as top element and $f = -\infty$ as bottom element. Please note that we slightly abuse notation here by using the same order symbol for functions as for the natural order on the reals.

Given an interpretation $\mathfrak{I} : \mathcal{X} \to \mathcal{F}$, a variable $X \in \mathcal{X}$, and a function $f \in \mathcal{F}$, we denote by $\mathfrak{I}[X \leftarrow f]$ the interpretation $\mathfrak{I}'$, such that $\mathfrak{I}'(X) = f$ and $\mathfrak{I}'(Y) = \mathfrak{I}(Y)$ for all $Y \neq X$.

Definition 2.4. Given a QTS $\mathcal{Q} = (V, E, \{P_i\}_{i\in I})$ and an interpretation $\mathfrak{I}$, a Qμ-formula yields a valuation function $[\![\varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}} : V \to \mathbb{R}_\infty$ defined as follows, for $v \in V$.

(1) $[\![P_i + c]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v) = P_i(v) + c$,

(2) $[\![X]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v) = \mathfrak{I}(X)(v)$,

(3) $[\![\neg\varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v) = -[\![\varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v)$,

(4) $[\![\varphi \wedge \psi]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v) = \min\{[\![\varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v), [\![\psi]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v)\}$,
$[\![\varphi \vee \psi]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v) = \max\{[\![\varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v), [\![\psi]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v)\}$,

(5) $[\![\Diamond\varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v) = \sup_{v' \in vE}[\![\varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v')$,
$[\![\Box\varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v) = \inf_{v' \in vE}[\![\varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v')$,

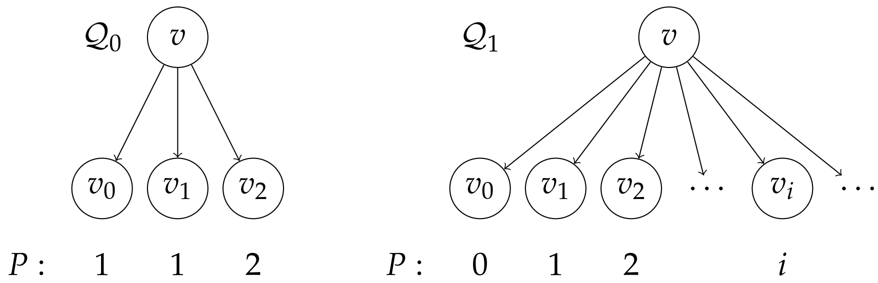


Figure 2.2: Evaluation of modal operators

(6) $[\![\mu X.\varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v) = \inf\{f \in \mathcal{F} : f = [\![\varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}[X \leftarrow f]}\}(v)$,
$[\![\nu X.\varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v) = \sup\{f \in \mathcal{F} : f = [\![\varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}[X \leftarrow f]}\}(v)$.

We extend the addition from $\mathbb{R}$ to $\mathbb{R}_\infty$ in the following way, let $a, b \in \mathbb{R}_\infty$ and we assume without loss of generality that $a \geq b$, then we define

$$a + b = \begin{cases} a + b & \text{if} \quad a \in \mathbb{R} \text{ and } b \in \mathbb{R} \\ \infty & \text{if} \quad a = \infty \text{ and } b > -\infty \\ -\infty & \text{if} \quad a < \infty \text{ and } b = -\infty \\ 0 & \text{if} \quad a = \infty \text{ and } b = -\infty. \end{cases}$$

The requirement that the fixed-point variable X occur only positively in the definition of a fixed-point formula and the monotonicity of the Boolean and modal operators guarantee the existence of least and greatest fixed points via the Knaster-Tarski theorem.

As in classical logic, we call a fixed-point variable *bound*, if it appears under the scope of a fixed-point operator. A variable that is not bound is called *free*. We call formulae without free variables *closed*, and we can simply write $[\![\varphi]\!]^{\mathcal{Q}}$ rather than $[\![\varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}}$ in this case.

We call the fragment of Qμ consisting of formulae without fixed-point operators *quantitative modal logic* QML.

If Qμ is interpreted over qualitative transition systems, it coincides with the classical μ-calculus and we say that $(\mathcal{Q}, v)$ is a model of φ, written $(\mathcal{Q}, v) \models \varphi$ if $[\![\varphi]\!]^{\mathcal{Q}}(v) = \infty$.

For a predicate symbol P_i and $c \in \mathbb{R}_\infty$, we use $|P_i - c|$ as an abbreviation for $(P_i + (-c)) \vee \neg(P_i + (-c))$.

Example 2.5. In Figure 2.2, we see two simple quantitative transition systems, a finite one, $\mathcal{Q}_0$, and an infinite one, $\mathcal{Q}_1$. Both have only a single quantitative predicate P and we depicted the values it assigns to each node below the respective node. Please note that $\mathcal{Q}_1$ is infinitely-branching, the node v has ω-many successors, with $P(v_i) = i$ respectively.
Let us look at two formulae, $\varphi = \Box(P+5)$ and $\psi = \Diamond P$ and illustrate the evaluation of modal operators on the two transition systems on the initial node v. On the first transition system, since the node v has only finitely many successors, the box operator simply evaluates to the minimum of the values of the succeeding atomic formula $(P+5)$ and hence we have $[\![\Box(P+5)]\!]^{\mathcal{Q}_0}(v) = 6$. The same holds true for ψ, it evaluates to the maximum of the successor values of v, and thus $[\![\Diamond P]\!]^{\mathcal{Q}_0}(v) = 2$
On the second transition system, we have infinitely many successors of the initial node. Box evaluates to the infimum, which in this case is also the minimum, plus 5, $[\![\Box(P+5)]\!]^{\mathcal{Q}_1}(v) = 5$. Diamond evaluates to the supremum of the successor values, thus, we have $[\![\Diamond P]\!]^{\mathcal{Q}_1}(v) = \infty$.

We always assume the formulae to be *well-named*, i.e. each fixed-point variable is bound only once and no variable appears both free and bound and we use the following common notions as defined e.g. in [24].

We say that a well-named formula φ of Qμ is in *negation normal form* if negations are only applied to atoms. We observe that, as for the classical μ-calculus, we can transform every well-named formula φ of Qμ into negation normal form by exploiting that $\neg\neg\varphi = \varphi$ and the following dualities.

- $\neg(\varphi \land \psi) \equiv \neg\varphi \lor \neg\psi$ and $\neg(\varphi \lor \psi) \equiv \neg\varphi \land \neg\psi$ (DeMorgan),
- $\neg\Box\varphi \equiv \Diamond\neg\varphi$ and $\neg\Diamond\varphi \equiv \Box\neg\varphi$,
- $\neg\mu X.\varphi \equiv \nu X.\neg\varphi[X/\neg X]$ and $\neg\nu X.\varphi \equiv \mu X.\neg\varphi[X/\neg X]$,

where $\varphi[X/\neg X]$ is identical to φ except that every occurrence of the fixed-point variable X is replaced by $\neg X$.

Furthermore, we call $\mathcal{D}_\varphi(X)$ the unique subformula in φ of the form $\mathrm{fp}X.\psi(X)$ where fp stands for either ν or μ. We assume every fixed-point variable X only occurs inside the scope of a quantifier and say that X' depends on X if X occurs free in $\mathcal{D}_\varphi(X')$. We call the transitive closure of this dependency relation the *dependency order* $<_\varphi$. Furthermore, for a variable X, we look at the variables it depends on and the induced $<_\varphi$-paths. We count the number of alternations of least and greatest fixed-point variables on all $<_\varphi$-paths and denote the

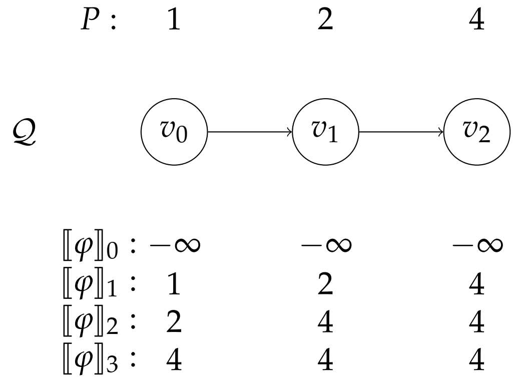


Figure 2.3: Inductive evaluation of $\varphi = \mu X.(P \vee \Diamond X)$

maximum by $\mathrm{al}_\varphi(X)$, the *alternation level* of the fixed-point variable X. Finally, we denote by $\mathrm{ad}(\varphi)$, the *alternation depth*, the maximal alternation level of all fixed-point variables occurring in φ.

Note that all operators in Qμ are monotone, thus guaranteeing the existence of the least and greatest fixed points. Furthermore, the fixed-points can be computed inductively according to the Knaster-Tarski Theorem stated below.

Proposition 2.6. *For* Qμ *formulae* $\mu X.\varphi$ *and* $\nu X.\varphi$ *built as described in Definition 2.3, the least and greatest fixed points exist and can be computed inductively:* $[\![\mu X.\varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}} = g_\gamma$ *with* $g_0(v) = -\infty$ *(and* $[\![\nu X.\varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}} = g_\gamma$ *with* $g_0(v) = \infty$*) for all* $v \in V$ *where*

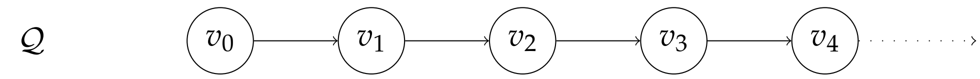
$$g_\alpha = \begin{cases} [\![\varphi]\!]_{\mathfrak{I}[X \leftarrow g_{\alpha-1}]} & \text{if } \alpha \text{ is a successor ordinal,} \\ \lim_{\beta<\alpha} [\![\varphi]\!]_{\mathfrak{I}[X \leftarrow g_\beta]} & \text{if } \alpha \text{ is a limit ordinal,} \end{cases}$$

and γ *is such that* $g_\gamma = g_{\gamma+1}$.

We call the smallest ordinal γ such that $g_\gamma = g_{\gamma+1}$ the *closure ordinal* of $[\![\mathrm{fp}X.\varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}}$ where $\mathrm{fp} \in \{\mu, \nu\}$.

Example 2.7. In Figure 2.3, we illustrate the inductive evaluation of a formula involving a least fixed point. The finite quantitative transition system $\mathcal{Q}$ consists of three nodes and has only a single quantitative predicate P. We consider the fixed-point formula $\varphi = \mu X.(P \vee \Diamond X)$ and evaluate it inductively according to Proposition 2.6. Since it is a least fixed point, we have to start at the bottom of the lattice with the function that assigns $-\infty$ to each node. Then, we

P_0 :	1	2	4	8	16
P_1 :	100	50	25	12.5	6.25



	$[\![\varphi]\!]_0$:	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$
	$[\![\varphi]\!]_1$:	1	2	4	8	16
	$[\![\varphi]\!]_2$:	2	4	8	12.5	16
	$[\![\varphi]\!]_3$:	4	8	12.5	12.5	16
	$[\![\varphi]\!]_4$:	4	12.5	12.5	12.5	16
$[\![\varphi]\!]_\infty =$	$[\![\varphi]\!]_5$:	12.5	12.5	12.5	12.5	16

Figure 2.4: Evaluation of $\varphi = \mu X.(P_0 \vee (P_1 \wedge \Diamond X))$

proceed through the stages of the evaluation as below. In Figure 2.3, we also depicted the values that each stage interpretation assigns to the nodes. For all $v \in V$,

$$\begin{aligned}
&[\![\varphi]\!]_0(v) = -\infty \\
&[\![\varphi]\!]_1(v) = P(v) \\
&[\![\varphi]\!]_2(v) = \max\{P(v), \sup_{w \in vE}\{[\![\varphi]\!]_1(w)\} \\
&[\![\varphi]\!]_3(v) = \max\{P(v), \sup_{w \in vE}\{[\![\varphi]\!]_2(w)\}\} = [\![\varphi]\!]_4(v) = [\![\varphi]\!]_\infty(v).
\end{aligned}$$

The fixed point is reached after four iterations. The classical meaning of the formula is that P is reachable from a node at which the formula holds. Here, the formula assigns to each node the highest value of P that is reachable from this node.

Example 2.8. Now we want to show another example of a fixed point evaluation, this time on an infinite quantitative transition system $\mathcal{Q}$, depicted in Figure 2.4. This transition system is basically just an infinite path. It has two quantitative predicates P_0 and P_1, where the values of P_0 get smaller along the path, while the values of P_1 get bigger. Formally, $P_0(v_i) = \frac{100}{2^i}$ and $P_1(v_i) = 2^i$. We now want to show how a classical Until formula evaluates in the quantitative setting. As we have explained in Section 1.1, a temporal formula P_1 Until P_0 holds on a path, if either P_0 holds immediately on the initial node, or it holds at a later node and on all nodes in between the property P_1 is true. We rewrite this temporal formula into the μ-calculus in the usual way and see what happens, if we evaluate it on the depicted transition system,

$$P_1 \text{ Until } P_0 := \ \mu X.(P_0 \vee (P_1 \wedge \Diamond X)).$$

As before, we are dealing with a least fixed-point formula and so we go through the stage evaluation again as detailed below. The values of the stages at each node are depicted in the figure.

$$\begin{aligned}
[\![\varphi]\!]_0(v) &= -\infty \\
[\![\varphi]\!]_1(v) &= \max\{P_0(v), \min\{P_1(v), \max_{w \in vE}\{\varphi_0(w)\}\} = P_0(v) \\
[\![\varphi]\!]_2(v) &= \max\{P_0(v), \min\{P_1(v), \max_{w \in vE}\{\varphi_1(w)\}\} \\
[\![\varphi]\!]_3(v) &= \dots
\end{aligned}$$

In the figure, we see that after six iterations the fixed point is reached. Intuitively, this formula evaluates to the biggest reachable value of P_1 at a node where $P_0 \leq P_1$ – if such a node is reachable. This holds for $\{v_0, v_1, v_2, v_3\}$ in this example and so they all evaluate to 12.5. If no such node is reachable, and thus, we have $P_0 > P_1$ for all reachable nodes – this holds for all v_i for $i \geq 4$ – then the formula just evaluates to the value of P_0.

Quantitative Modal Logic

As mentioned before, the fragment of Qμ without the use of fixed-point operators is called quantitative modal logic QML. In Section 2.4 below, we also need the following notions.

Definition 2.9. The *modal nesting depth* dp of a formula of QML is defined as follows:

- $\mathrm{dp}(P_i + c) = 0$ for a predicate P_i and $c \in \mathbb{R}_\infty$,
- $\mathrm{dp}(\neg\varphi) = \mathrm{dp}(\varphi)$,
- $\mathrm{dp}(\varphi \vee \psi) = \mathrm{dp}(\varphi \wedge \psi) = \max\{\mathrm{dp}(\varphi), \mathrm{dp}(\psi)\}$,
- $\mathrm{dp}(\Box\varphi) = \mathrm{dp}(\Diamond\psi) = 1 + \mathrm{dp}(\varphi)$.

We define the fragment QML^n of QML by allowing only formulae with a modal nesting depth at most n. We define *logical equivalence up to a real number* r (or ∞), meaning that the evaluations of all formulae of a quantitative logic differ by at most r.

Definition 2.10. For a logic $\mathcal{L}$, we write that $(\mathcal{Q}, v) \equiv^r_{\mathcal{L}} (\mathcal{Q}', v')$, if for all $\varphi \in \mathcal{L}$ we have that $|[\![\varphi]\!]^{\mathcal{Q}}(v) - [\![\varphi]\!]^{\mathcal{Q}'}(v')| \leq r$.

We use this notion for both Qμ and QML, denoted by $\equiv^r_{\mathrm{QML}}$ and $\equiv^r_{\mathrm{Q}\mu}$.

2.2 Bisimulation Relation and Games

We introduce quantitative bisimulation and show that, as in the classical case, our quantitative μ-calculus is invariant with respect to quantitative bisimulation. Obviously, the classical definition of a bisimulation relation is far too restrictive in the quantitative setting. Instead, we take an approach similar to [10, 40] and define a bisimulation distance that intuitively tells us how much two given systems differ from each other. The concept of invariance is subsequently also replaced by a notion of differences in evaluations of formulae.

The motivation for defining quantitative bisimulation and invariance in such a way stems from practical application scenarios. One might easily imagine that, in practice, minor variations in the quantitative predicates should also only slightly influence the evaluation of a formula. The fact that a requirement is fulfilled should be robust under minor perturbations of the values.

First, let us look at an example.

Example 2.11. Figure 2.5 depicts three quantitative transition systems with the same underlying graph. They only differ in the values of the quantitative predicate P. As classical models, these systems are not bisimilar. They have different values at the nodes and this would be modelled by different predicates. However, in the quantitative setting, we see that the first two systems, $\mathcal{Q}_0$ and $\mathcal{Q}_1$ are much more similar than $\mathcal{Q}_0$ and $\mathcal{Q}_2$, as their values only differ by 0.1 whereas, the difference between $\mathcal{Q}_0$ and $\mathcal{Q}_2$ is 99.

A quantitative relation or distance should formalise this intuition and give us exactly these differences as the maximum (supremum) of all the differences in the predicate evaluations. Additionally, it should generalise classical bisimulation.

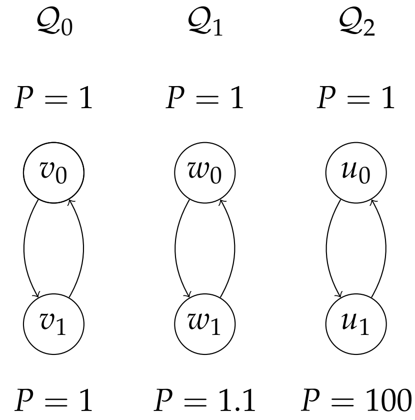


Figure 2.5: How different are these systems?

First, we define a distance between states of different transition systems as proposed in [10]. For this purpose, we compare all values of propositions at two states and take the maximum of the differences.

Definition 2.12. ([10]) The *propositional distance* between two states is the maximal distance in their proposition evaluations, i.e. $\text{pd} : V \times V' \to \mathbb{R}^+_\infty$ is defined by,

$$\text{pd}(v, w) = \max_{i \in I}\{|(P_i(v) - P_i(w))|\}.$$

In [10, 40], bisimulation distance is introduced as a least fixed point of an operator on functions on tuples of states equipped with a distance function. Additionally, a coalgebraic definition is given in [40]. In contrast, we define bisimulation as a relation and as a game. A natural way to define a quantitative bisimulation, i.e. a bisimulation distance, is to consider bisimulation up to a real value r as in the definition below.

Definition 2.13. A *quantitative bisimulation relation* $\approx^r$ with $r \in \mathbb{R}^+_0$ between two transition systems $\mathcal{Q} = (V, E, \{P_i\}_{i \in I})$ and $\mathcal{Q}' = (V', E', \{P'_i\}_{i \in I})$ is a non-empty relation $\approx^r \subseteq V \times V'$, such that for all $(v, v') \in \approx^r$,

– $\text{pd}(v, v') \leq r$ (propositional distance),

– for every $w \in vE$, there is a $w' \in v'E'$ with $(w, w') \in \approx^r$ (forth condition),

– for every $w' \in v'E'$, there is a $w \in vE$ with $(w, w') \in \approx^r$ (back condition).

We call two pointed quantitative transition systems, $(\mathcal{Q}, v)$ and $(\mathcal{Q}', v')$ *r-bisimilar* if there exists a quantitative bisimulation relation $\approx^r$ and $(v, v') \in \approx^r$, and we simply write $(\mathcal{Q}, v) \approx^r (\mathcal{Q}', v')$. If there is no $r \in \mathbb{R}_0^+$, such that there exists a relation $\approx^r$, then $(\mathcal{Q}, v)$ and $(\mathcal{Q}', v')$ are not bisimilar denoted by $(\mathcal{Q}, v) \not\approx (\mathcal{Q}', v')$. For qualitative transition systems, classical bisimulation corresponds to $\approx^0$, meaning that two qualitative transition systems $(\mathcal{Q}, v)$ and $(\mathcal{Q}', v')$ are bisimilar in the classical sense if there exists a relation $\approx^0$, such that $(\mathcal{Q}, v) \approx^0 (\mathcal{Q}', v')$ and they are not bisimilar if no such relation exists.

In the classical setting, the equivalence of structures with respect to a logic is also treated with game-theoretic methods, e.g. with the help of Ehrenfeucht-Fraïssé games, model comparison games for first-order logic (see e.g. [28, 37]). These games are two-player games, where one player tries to show that two systems are indistinguishable for a logic, whereas the other player wants to prove the opposite. Classical Ehrenfeucht-Fraïssé games are used to decide whether two structures are elementary equivalent, i.e. indistinguishable for first-order logic. They are also often used to show non-definability in first-order logic and work for general models as well as in finite model theory, a remarkable feature as classical concepts often do not easily translate to finite models.

As classical modal logic can be seen as a fragment of first-order logic, it also has a variant of Ehrenfeucht-Fraïssé games that capture the notion of bisimulation. As we already stated in the introduction, however, bisimulation does not exactly correspond to equivalence in modal logic, only on the restriction to finitely-branching systems. This game characterisation of bisimulation has also been studied independently, i.e. without the connection to modal logic. Bisimulation games have some advantages over the recursive definition as a relation. They are intuitive and easy to understand and most importantly pave the way for a game-based analysis. In fact, many of the classical proofs relating bisimulation to modal equivalence make heavy use of the game-based description of bisimulation. We refer the reader to [23] for an extensive treatment of classical bisimulation results.

In the quantitative world, these advantages may be even more obvious. First of all, it is easy to adapt the classical bisimulation game and design a quantitative game that captures our intuition of what a quantitative bisimulation relation should accomplish. This is again a two-player game, one player, Spoiler, wants to maximise the difference between two systems whereas the other player, Duplicator, wants to minimise it. The value of this game defines the bisimulation distance. We show that using this game-based description, we can adapt the classical proofs and lift the corresponding model-theoretic theorems to the quantitative setting.

First, we give a formal definition of the bisimulation game. Then, we show that, as in the classical case, the game definition is equivalent to the recursive definition.

Definition 2.14. For two quantitative transition systems $\mathcal{Q}_0, \mathcal{Q}_1$ we define the *bisimulation game*,

$$\mathcal{G}_\sim(\mathcal{Q}_0, v, \mathcal{Q}_1, v')$$

as follows. There are two players, Spoiler and Duplicator. At the beginning of a play, a token is placed on both transition systems on v and v'. In every round, if the token is currently on (u_0, u_1),

– Spoiler chooses a transition system $\mathcal{Q}_i, i \in \{0,1\}$ and moves the token to a node $w_i \in u_i E_i$.

– Duplicator now moves the token in $\mathcal{Q}_{1-i}$ to a $w_{1-i} \in u_{1-i} E_{1-i}$.

The play ends if one of the players cannot move anymore. If Spoiler cannot move anymore, the payoff will be calculated as below, but if Duplicator cannot respond to a move by Spoiler because the current state she is in has no successors, the play has a payoff of ∞. Duplicator wants to minimise the outcome and Spoiler wants to maximise it.

The *outcome* of a play $\pi = (v_0, v'_0), (v_1, v'_1), \ldots$ is

$$\mathrm{p}(\pi) = \sup_{(v,v') \in \pi} \mathrm{pd}(v, v').$$

A *strategy* γ *of Spoiler* is a function, that assigns to a history

$$(v_0, v'_0), \ldots, (v_i, v'_i) \in (V \times V')^{i+1}$$

a node $v \in V$ (or $v' \in V'$), such that $(v_i, v) \in E$ (or $(v'_i, v') \in E'$).

A *strategy* β *of Duplicator* is a function, that assigns to a history

$$(v_0, v'_0), \ldots, (v_{i-1}, v'_{i-1}), (v_i, \_) \text{ or } (v_0, v'_0), \ldots, (v_{i-1}, v'_{i-1}), (\_, v'_i)$$

a node $v \in V$ or $v' \in V'$, such that $(v_{i-1}, v_i) \in E$ or $(v'_{i-1}, v'_i) \in E'$.

A play π starting from (v_0, v'_0) is *consistent with a strategy* γ *of Spoiler* if for all play prefixes of π, $(v_0, v'_0), \ldots, (v_i, v'_i), (v_{i+1}, v'_{i+1})$, we have one of the following.

- $\gamma((v_0, v'_0), \ldots, (v_i, v'_i)) = v_{i+1}$ or
- $\gamma((v_0, v'_0), \ldots, (v_i, v'_i)) = v'_{i+1}$.

A play π starting from (v_0, v'_0) is *consistent with a strategy β of Duplicator* if for all play prefixes of π, $(v_0, v'_0), \ldots, (v_i, v'_i), (v_{i+1}, v'_{i+1})$, we have one of the following.

- $\beta((v_0, v'_0), \ldots, (v_i, v'_i), (v_{i+1}, \_)) = v'_{i+1}$ or
- $\beta((v_0, v'_0), \ldots, (v_i, v'_i), (\_, v'_{i+1})) = v_{i+1}$.

The unique play starting from (v, v') consistent with strategies γ and β is denoted by $\pi_{\beta,\gamma}(v, v')$.

A bisimulation game $\mathcal{G}_\sim(\mathcal{Q}, v, \mathcal{Q}', v')$ is *determined* if

$$\inf_{\beta \in \Gamma_2} \sup_{\gamma \in \Gamma_1} \mathrm{p}(\pi_{\beta,\gamma}(v, v')) = \sup_{\gamma \in \Gamma_1} \inf_{\beta \in \Gamma_2} \mathrm{p}(\pi_{\beta,\gamma}(v, v')) =: \mathrm{val}\mathcal{G}_\sim(\mathcal{Q}, v, \mathcal{Q}', v'),$$

where Γ_1 and Γ_2 denote the sets of strategies for Spoiler and Duplicator. We call $\mathrm{val}\mathcal{G}_\sim(\mathcal{Q}, v, \mathcal{Q}', v')$ the *value of the bisimulation game*. If $\mathrm{val}\mathcal{G}_\sim(\mathcal{Q}, v, \mathcal{Q}', v') = r$, we write $(\mathcal{Q}, v) \sim^r (\mathcal{Q}', v')$.

The *n-round bisimulation game* $\mathcal{G}^n_\sim(\mathcal{Q}, v, \mathcal{Q}', v')$ is played in the same way as the full bisimulation game, except it ends after n rounds (for $n \in \mathbb{N}$) - or earlier if one of the players cannot move anymore. The payoff is computed exactly in the same way as for the full game. If $\mathrm{val}\mathcal{G}^n_\sim(\mathcal{Q}, v, \mathcal{Q}', v') = r$, we write $\mathcal{Q}, v \sim^r_n \mathcal{Q}', v'$.

To avoid confusion when referring to the players by pronouns, we regard Spoiler as male and Duplicator as female and refer to them as "he" and "she".

A strategy is *positional*, if it chooses the next move only according to the current position in the game and not regarding the history. Formally, a positional strategy β of Duplicator is a function, that assigns to a position

$$((v_{i-1}, v'_{i-1}), (v_i, \_)) \text{ or } ((v_{i-1}, v'_{i-1}), (\_, v'_i))$$

a node $v \in V$ or $v' \in V'$ such that $(v_{i-1}, v_i) \in E$ or $(v'_{i-1}, v'_i) \in E'$.

Next, we show that bisimulation games are determined and therefore the value is well-defined. For a given value r, a bisimulation game can be seen as a safety game for the Duplicator. Her winning objective in this game is to stay in the region where the propositional distance does not exceed r.

Formally, a *safety game* is a graph game between two players where the positions are assigned one of two colours. The safety condition is described by the target colour. Player 0 wins a play in a safety game if all the positions in the play are of this colour. It is well-known that safety games are *positionally determined*, i.e. each position belongs to the winning region of one of the players and if Player 0 has a winning strategy from a position, she also has a positional one. We now show that the same property holds for bisimulation games.

First of all, we deduce the determinacy of bisimulation games as follows. To formalise the above intuition, for each $r \in \mathbb{R}$ we can re-phrase the bisimulation game on $\mathcal{Q}$ and $\mathcal{Q}'$ up to r as a safety game: all positions, i.e. tuples (v, v'), $(v, \_)$, or $(\_, v')$, are coloured in the following way. Positions $(v, \_)$ or $(\_, v')$ get colour 0. For positions (v, v'), if their propositional distance $\mathrm{pd}(v, v') \leq r$ they get colour 0, if $\mathrm{pd}(v, v') > r$ they get colour 1. The safety condition for Duplicator then is to stay within 0-coloured positions. As safety games are determined, all we have to do to determine the value of $\mathcal{G}_\sim(\mathcal{Q}, v, \mathcal{Q}', v')$ is to take the infimum over all r such that Duplicator wins the safety game up to r. Furthermore, we get positional determinacy utilising the following proposition.

Proposition 2.15. *Fix two quantitative transition systems $\mathcal{Q}$ and $\mathcal{Q}'$. For every $r \in \mathbb{R}$ there exists a positional strategy β such that if $\mathrm{val}\mathcal{G}_\sim(\mathcal{Q}, v, \mathcal{Q}', v') \leq r$ then for every strategy γ of Spoiler we have $\mathrm{p}(\pi_{\beta,\gamma}) \leq r$*

Proof. We again re-phrase the bisimulation game $\mathcal{G}_\sim(\mathcal{Q}, v, \mathcal{Q}', v')$ up to a fixed r as a safety game (as above). If $\mathrm{val}\mathcal{G}_\sim(\mathcal{Q}, v, \mathcal{Q}', v') \leq r$ then Duplicator has a strategy that guarantees her that she never visits a position (v, v') with $\mathrm{pd}(v, v') > r$, as then the value would be greater than r as well. But this is a winning strategy in the safety game for $\mathcal{G}$. Safety games are positionally determined, this means that if Player 0 has a winning strategy, she has a positional one. Thus, Duplicator has a positional strategy β to guarantee that she never visits a position (v, v') with $\mathrm{pd}(v, v') > r$. This is also a positional strategy in the original game to assure an outcome at most r. $\square$

Having established positional determinacy for bisimulation games, let us now look at an example.

Example 2.16. Figure 2.6 depicts two quantitative transition systems $\mathcal{K}$ and $\mathcal{H}$. Both have only one quantitative predicate P. The values of P at each node are depicted next to it. Also, we notice that we can make two consecutive moves

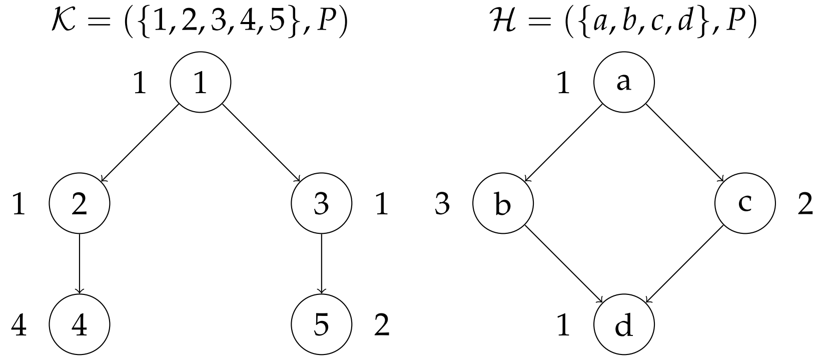


Figure 2.6: Two systems $(\mathcal{K}, 1)$ and $(\mathcal{H}, a)$ that are $\sim_3$-bisimilar

in each system. Spoiler will begin the game and, as he wants to maximise the occurring differences in the values of P, following an optimal strategy he will first move in $\mathcal{K}$ from 1 to 2. As Duplicator wants to minimise, she will move in $\mathcal{H}$ from a to c. Spoiler challenges again with a move from 2 to 4. Duplicator has to reply with a move from c to d, resulting in the play $\pi = (1,a),(2,c),(4,d)$ with the outcome $\sup_{(v,v')\in\pi} \mathrm{pd}(v,v') = \sup\{0,2,3\} = 3$. This means that the systems are $\sim_3$-bisimilar, $(\mathcal{K},1) \sim_3 (\mathcal{H},a)$.

The relational and the game description of r-bisimulation are equivalent as stated in the following proposition.

Proposition 2.17. *For two pointed quantitative transition systems $(\mathcal{Q},v),(\mathcal{Q}',v')$, we have that* $\mathrm{val}\mathcal{G}_\sim(\mathcal{Q},v,\mathcal{Q}',v') \le r$*, i.e.* $\mathcal{Q},v \sim^r \mathcal{Q}',v'$*, if and only if* $\mathcal{Q},v \approx^r \mathcal{Q}',v'$.

Proof. $(\Rightarrow)$
If $\mathrm{val}\mathcal{G}_\sim(\mathcal{Q},v,\mathcal{Q}',v') = r$, then Duplicator has a strategy to guarantee an outcome at most r from (v,v'). Let M be the set of all tuples (w,w'), such that Duplicator has a strategy to guarantee an outcome at most r from this position, $M = \{(w,w') |\ \mathrm{val}\mathcal{G}_\sim(\mathcal{Q},w,\mathcal{Q}',w') \le r\}$. We have to show that M is an r-bisimulation relation.
First, we note that it is non-empty since it contains at least the tuple (v,v') by assumption. Furthermore, we observe that trivially $\mathrm{pd}(w,w') \le r$ for every $(w,w') \in M$, since otherwise $\mathrm{val}\mathcal{G}_\sim(\mathcal{Q},w,\mathcal{Q}',w') > r$ and this only leaves us to check the back and forth conditions from Definition 2.13. Let β be her positional strategy (from Proposition 2.15). The forth condition states that for all $(w,w') \in M$ we have a $u' \in w'E'$ for every $u \in wE$, such that $(u,u') \in M$. Let

us check that this holds. Let $(w, w') \in M$. For every challenge $u \in wE$ or in game terminology for a position $(w, w'), (u, \_)$ with $u \in wE$, Duplicator has a response $u' \in w'E'$ with $\beta((w, w'), (u, \_)) = u'$. Since $\mathrm{val}\mathcal{G}_{\sim}(\mathcal{Q}, w, \mathcal{Q}', w') \leq r$ by assumption and she plays according to β, also $\mathrm{val}\mathcal{G}_{\sim}(\mathcal{Q}, u, \mathcal{Q}', u') \leq r$ and thus $(u, u') \in M$. The back condition is proved analogously, just that now the challenge is in $\mathcal{Q}$ but a strategy for Duplicator gives her an appropriate answer as well. This means that M is an r-bisimulation relation $\approx^r$.

$(\Leftarrow)$

The relation $\approx^r$ provides a non-deterministic strategy for Duplicator to get an outcome at most r. By assumption, the initial position $(v, v') \in \approx^r$. This also guarantees that she does not immediately get an outcome greater than r as for every $(v, v') \in \approx^r$ we have that $\mathrm{pd}(v, v') \leq r$ and the value of the bisimulation game is calculated using the supremum of propositional distances occurring in a play. Assume that in a play we have for a tuple $(w, w') \in \approx^r$ a challenge $(w, w'), (u, \_)$ of Spoiler. Duplicator can just choose any $u' \in w'E'$ with $(u, u') \in \approx^r$. Such a tuple exists by the forth condition. Analogously, for every challenge in the other system, the back condition guarantees her the existence of an appropriate answer in $\approx^r$. As Duplicator can now force the whole play π to stay within $\approx^r$, this means that for every $(u, u') \in \pi$ we have by definition that $\mathrm{pd}(u, u') \leq r$ and thus also $\mathrm{p}(\pi) = \sup_{(u,u') \in \pi} \mathrm{pd}(u, u') \leq r$. Thus, $val\mathcal{G}_{\sim}(\mathcal{Q}, v, \mathcal{Q}', v') \leq r$ or $\mathcal{Q}, v \sim^r \mathcal{Q}', v'$. □

2.3 Bisimulation Invariance

In the classical setting, a logic is invariant under a relation if no formula of the logic can distinguish between structures that are related. The modal μ-calculus, and thus modal logic, is invariant under bisimulation, no formula of the modal μ-calculus can distinguish between bisimilar states.

In the quantitative world, we have defined bisimulation up to a real value r. Hence, the invariance theorem will have the following shape: if two structures differ by at most r (are r-bisimilar) then also the evaluation of all formulae will differ by at most r. As stated before, the reverse direction is not true for the μ-calculus. However, for finitely-branching systems, modal logic characterises bisimulation equivalence. This means that if finitely-branching structures are bisimilar then they are equivalent for modal logic.

In the following section, we show that the corresponding results also hold in the quantitative world.

The invariance theorem can be proved by an induction on the structure of the formula. Later, we will present an alternative way to prove this theorem using another game concept – model-checking games (see Section 3.4).

Theorem 2.18. *Let $\varphi \in \mathrm{Q}\mu$ and $v \in \mathcal{Q}, v' \in \mathcal{Q}'$ with $\mathcal{Q}, v \approx^r \mathcal{Q}', v'$ then $|[\![\varphi]\!]^{\mathcal{Q}}(v) - [\![\varphi]\!]^{\mathcal{Q}'}(v')| \leq r$ or, in other words,*

$$(\mathcal{Q}, v) \approx^r (\mathcal{Q}', v') \text{ implies } (\mathcal{Q}, v) \equiv^r_{\mathrm{Q}\mu} (\mathcal{Q}', v').$$

Proof. Assume we have two pointed quantitative transition systems, $(\mathcal{Q}, v)$ and $(\mathcal{Q}', v')$, and we know that $(\mathcal{Q}, v) \approx^r (\mathcal{Q}', v')$. We proceed by induction on the structure of the formula φ.

Base case: $\varphi = P_i + c$ for some P_i and constant $c \in \mathbb{R}$. Since $(\mathcal{Q}, v) \approx^r (\mathcal{Q}', v')$ we know that $\mathrm{pd}(v, v') = \max_{i \in I} |P_i(v) - P_i(v')| \leq r$. Hence, for all $i \in I$, we have $|P_i(v) - P_i(v')| \leq r$ and therefore also $|P_i(v) + c - (P_i(v') + c)| \leq r$.

For the induction step, it suffices to consider the cases $\neg\varphi$, $\varphi \vee \psi$, $\Diamond\varphi$ and $\nu X.\varphi(X)$.

First case: $\varphi = \neg\psi$.
We need to show that $|[\![\neg\psi]\!]^{\mathcal{Q}}(v) - [\![\neg\psi]\!]^{\mathcal{Q}'}(v')| \leq r$. We have that

$$\begin{aligned} & |[\![\neg\psi]\!]^{\mathcal{Q}}(v) - [\![\neg\psi]\!]^{\mathcal{Q}'}(v')| \\ =& |-[\![\psi]\!]^{\mathcal{Q}}(v) - (-[\![\psi]\!]^{\mathcal{Q}'}(v'))| \\ =& |[\![\psi]\!]^{\mathcal{Q}'}(v') - [\![\psi]\!]^{\mathcal{Q}}(v))| \leq r, \end{aligned}$$

since by induction hypothesis $|[\![\psi]\!]^{\mathcal{Q}}(v) - [\![\psi]\!]^{\mathcal{Q}'}(v'))| \leq r$.

Second case: $\psi = \psi \vee \vartheta$.
As we know that

$$\begin{aligned} & |[\![\psi \vee \vartheta]\!]^{\mathcal{Q}'}(v') - [\![\psi \vee \vartheta]\!]^{\mathcal{Q}}(v))| \\ =& |\max\{[\![\psi]\!]^{\mathcal{Q}}(v), [\![\vartheta]\!]^{\mathcal{Q}}(v)\} - \max\{[\![\psi]\!]^{\mathcal{Q}'}(v'), [\![\vartheta]\!]^{\mathcal{Q}'}(v')\}| \end{aligned}$$

by definition, we have to consider two cases. Either the formula with the maximal evaluation is equal for both nodes, let us say without loss of generality it is ψ. Then

$$\begin{aligned} & |\max\{[\![\psi]\!]^{\mathcal{Q}}(v), [\![\vartheta]\!]^{\mathcal{Q}}(v)\} - \max\{[\![\psi]\!]^{\mathcal{Q}'}(v'), [\![\vartheta]\!]^{\mathcal{Q}'}(v')\}| \\ =& |[\![\psi]\!]^{\mathcal{Q}}(v) - [\![\psi]\!]^{\mathcal{Q}'}(v')| \end{aligned}$$

which by induction hypothesis is smaller or equal to r.

Or it is different for the two systems, without loss of generality we assume $\max\{[\![\psi]\!]^{\mathcal{Q}}(v), [\![\vartheta]\!]^{\mathcal{Q}}(v)\} = [\![\psi]\!]^{\mathcal{Q}}(v)$ and $\max\{[\![\psi]\!]^{\mathcal{Q}'}(v'), [\![\vartheta]\!]^{\mathcal{Q}'}(v')\} = [\![\vartheta]\!]^{\mathcal{Q}'}(v')$. Then

$$|[\![\psi \vee \vartheta]\!]^{\mathcal{Q}'}(v') - [\![\psi \vee \vartheta]\!]^{\mathcal{Q}}(v))| = |[\![\psi]\!]^{\mathcal{Q}}(v) - [\![\vartheta]\!]^{\mathcal{Q}'}(v')|.$$

If $[\![\psi]\!]^{\mathcal{Q}}(v) > [\![\vartheta]\!]^{\mathcal{Q}'}(v')$ we use that by assumption $[\![\vartheta]\!]^{\mathcal{Q}'}(v') \geq [\![\psi]\!]^{\mathcal{Q}'}(v')$ and again by induction hypothesis $|[\![\psi]\!]^{\mathcal{Q}}(v) - [\![\psi]\!]^{\mathcal{Q}'}(v')| \leq r$.
Otherwise, $[\![\psi]\!]^{\mathcal{Q}}(v) \leq [\![\vartheta]\!]^{\mathcal{Q}'}(v')$, we also know that $[\![\vartheta]\!]^{\mathcal{Q}}(v) \leq [\![\psi]\!]^{\mathcal{Q}}(v)$ and again by induction hypothesis $|[\![\vartheta]\!]^{\mathcal{Q}}(v) - [\![\vartheta]\!]^{\mathcal{Q}'}(v')| \leq r$.
Altogether, we have $|[\![\psi]\!]^{\mathcal{Q}}(v) - [\![\vartheta]\!]^{\mathcal{Q}'}(v')| \leq r$.

Third case: $\varphi = \Diamond\psi$. Towards a contradiction, assume that

$$|[\![\Diamond\psi]\!]^{\mathcal{Q}}(v) - [\![\Diamond\psi]\!]^{\mathcal{Q}'}(v'))| = |\sup_{w \in vE}\{[\![\Diamond\psi]\!]^{\mathcal{Q}}(w)\} - \sup_{w' \in v'E'}\{[\![\Diamond\psi]\!]^{\mathcal{Q}'}(w')\}| > r.$$

Subcase 1: $[\![\Diamond\psi]\!]^{\mathcal{Q}}(v) \geq [\![\Diamond\psi]\!]^{\mathcal{Q}'}(v')$.
Let $w_{\max} \in vE$ such that $[\![\psi]\!]^{\mathcal{Q}}(w_{\max}) \geq \sup_{w' \in v'E}\{[\![\psi]\!]^{\mathcal{Q}'}(w')\} + r$. Please note that the case that both suprema are infinite is ruled out by our assumption. Since $(\mathcal{Q}, v) \approx^r (\mathcal{Q}', v')$, we know by the forth condition that for every $w \in vE$ there is a $w' \in v'E'$, such that $(\mathcal{Q}, w) \approx^r (\mathcal{Q}', w')$, this means in particular that we can find a $u' \in v'E'$, such that $(\mathcal{Q}, w_{\max}) \approx^r (\mathcal{Q}', u')$ which means by induction hypothesis that $|[\![\psi]\!]^{\mathcal{Q}}(w_{\max}) - [\![\psi]\!]^{\mathcal{Q}'}(u'))| \leq r$. But since $[\![\psi]\!]^{\mathcal{Q}'}(u') \leq \sup_{w' \in v'E'}\{[\![\psi]\!]^{\mathcal{Q}'}(w')\}$, it also follows that $|[\![\psi]\!]^{\mathcal{Q}}(w_{\max}) - [\![\psi]\!]^{\mathcal{Q}'}(u'))| > r$ which is a contradiction.

Subcase 2: $[\![\Diamond\psi]\!]^{\mathcal{Q}}(v) < [\![\Diamond\psi]\!]^{\mathcal{Q}'}(v')$. Analogously, let ${w'}_{\max} \in v'E'$ such that $[\![\psi]\!]^{\mathcal{Q}'}({w'}_{\max}) \geq \sup_{w \in vE}\{[\![\psi]\!]^{\mathcal{Q}}(w)\} + r$. Since $(\mathcal{Q}, v) \approx^r (\mathcal{Q}', v')$, we know by the back condition that for every $w' \in v'E'$ there is a $w \in vE$, such that $(\mathcal{Q}, w) \approx^r (\mathcal{Q}', w')$, this means in particular that we can find a $u \in vE$, such that $(\mathcal{Q}, u) \approx^r (\mathcal{Q}', {w'}_{\max})$ which means by induction hypothesis that $|[\![\Diamond\psi]\!]^{\mathcal{Q}}(u) - [\![\Diamond\psi]\!]^{\mathcal{Q}'}({w'}_{\max}))| \leq r$. But since $[\![\Diamond\psi]\!]^{\mathcal{Q}}(u) \leq \sup_{w \in vE}\{[\![\Diamond\psi]\!]^{\mathcal{Q}}(w)\}$, it also follows that $|[\![\Diamond\psi]\!]^{\mathcal{Q}}(u) - [\![\Diamond\psi]\!]^{\mathcal{Q}'}({w'}_{\max}))| > r$ which is again a contradiction.

Fourth case: $\varphi = \nu X.\psi(X)$.
Let $[\![\nu X.\psi(X)]\!]^{\mathcal{Q}} = g_\alpha$ and $[\![\nu X.\psi(X)]\!]^{\mathcal{Q}'} = g'_{\alpha'}$ for the respective closure ordinals α, α'. By the induction hypothesis, we have that

$$|[\![\psi[X \leftarrow g]]\!]^{\mathcal{Q}}(v) - [\![\psi[X \leftarrow g']]\!]^{\mathcal{Q}'}(v')| \leq r$$

for all interpretations g, g' of the fixed-point variable X such that $|g(v) - g'(v')| \leq r$. To see that this holds for g_γ and ${g'}_\gamma$ for $\gamma = \max\{\alpha, \alpha'\}$ we look at the stages of the evaluation. For $g_0(v) = g'_0(v') = \infty$ this trivially holds. For $g_\beta(v)$ and ${g'}_\beta(v')$ for a successor ordinal β, this holds by induction. For a limit ordinal β, we also have $|g_\beta(v) - {g'}_\beta(v')| \leq r$ because for all $\beta' < \beta$ we have by induction that $|g_{\beta'}(v) - g'_{\beta'}(v')| \leq r$. Thus, we have

$$|[\![\psi[X \leftarrow g_\gamma]]\!]^{\mathcal{Q}}(v) - [\![\psi[X \leftarrow g'_\gamma]]\!]^{\mathcal{Q}'}(v)| \leq r$$

and therefore

$$|[\![\nu X.\psi(X)]\!]^{\mathcal{Q}}(v) - [\![\nu X.\psi(X)]\!]^{\mathcal{Q}}(v')| \leq r.$$

□

2.4 Characteristic Formulae

In classical modal logic, characteristic formulae are used to describe a fixed transition system up to finitary bisimulation. In particular, these formulae formalise that fact that Duplicator has a winning strategy in the n-round bisimulation game. In the classical setting this means that, if the characteristic formula up to n for a system $(\mathcal{K}, v)$ holds for $(\mathcal{K}', v')$, then Duplicator has a winning strategy for the n-round bisimulation game played on $(\mathcal{K}, v)$ and $(\mathcal{K}', v')$.

In the quantitative world, there is also a way to construct characteristic formulae. In [40], characteristic formulae which are syntactically similar to the classical ones appear in the logical characterisation of bisimulation. The ones that we use, and that appear in [10], are also similar to the classical ones except that all operators are swapped for their dual ($\wedge$ for $\vee$, $\Box$ for $\Diamond$ and vice versa). This is due to the fact that these formulae should evaluate to the bisimulation distance, so in particular to 0 if the systems are classically bisimilar.

For a given pointed quantitative transition system $(\mathcal{Q}, v)$ with only finitely many quantitative predicates, we define the characteristic formula up to nesting depth n inductively.

$$\chi^0_{[\mathcal{Q},v]} = \bigvee_{P_i} |P_i - P_i(v)|,$$

$$\chi^{n+1}_{[\mathcal{Q},v]} = \chi^0_{[\mathcal{Q},v]} \vee \bigvee_{w \in vE} \Box \chi^n_{[\mathcal{Q},w]} \vee \Diamond \bigwedge_{w \in vE} \chi^n_{[\mathcal{Q},w]}.$$

As in the classical setting, the formula has a game-theoretic meaning (which is not discussed in [10, 40]). It describes that Duplicator can guarantee an outcome less than or equal to the value of the formula in the n-round bisimulation game. We can use this to prove the back direction of the invariance theorem and thus get a characterisation theorem.

Please note that – unlike in the classical case – we have to restrict the transition systems to be finitely-branching from the beginning and not just when we consider full bisimulation. In the classical case, if there are only finitely many predicates and finitely many different modal types, this guarantees that the disjunctions and conjunctions in the characteristic formulae are all finite as we have only finitely many non-equivalent formulae $\chi^n_{[\mathcal{Q},v]}$ at each level n. In the quantitative case, there could be infinitely many non-equivalent $\chi^n_{[\mathcal{Q},v]}$ in an infinitely-branching system. Imagine, e.g. a node that has infinitely many successors (as in Figure 2.2), one for each natural number and a quantitative predicate that assigns that number to the respective node. There would be infinitely many different non-equivalent formulae $\chi^0_{[\mathcal{Q},v]}$.

Please also note that we use addition of real constants for atomic formulae for constructing $\chi^0_{[\mathcal{Q},v]}$. The following example shows that this is crucial and allows us to distinguish between systems that are 1-bisimilar but cannot be distinguished without addition.

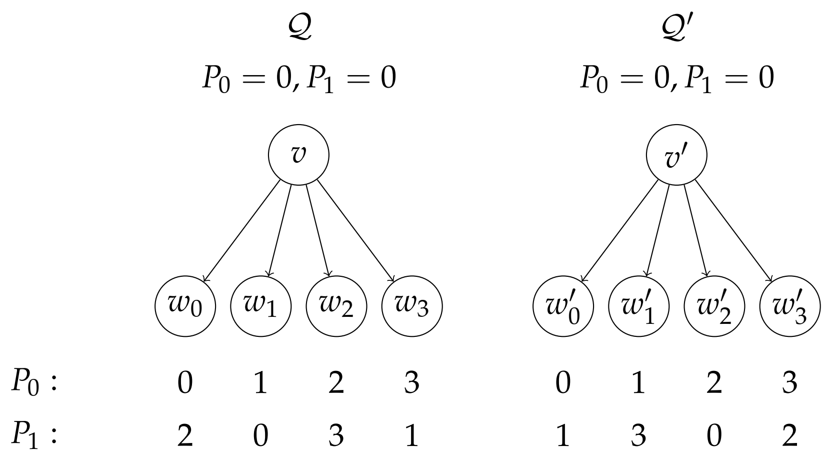


Figure 2.7: Two systems that cannot be distinguished without addition.

Example 2.19. In Figure 2.7, we illustrate by an example why we need to allow addition of real values in atomic formulae of Qμ and QML. These systems have two quantitative predicates, P_0 and P_1, their values are depicted in the figure. The value of the bisimulation game is 1, thus the systems are 1-bisimilar. Intuitively, the difficulty with pinpointing the difference between the systems with a formula is that we have the same values of quantitative predicates in both systems, only in different combinations. However, Qμ without addition of real constants in atomic formulae can only speak about a set of values in terms of minima and maxima, but this does not help us to distinguish the systems. The notion of propositional distance, however, allows to compare all values at once.

To see that full QML (and then also Qμ) can distinguish between these systems, let us build the characteristic formula for $(\mathcal{Q}, v)$:

$$\begin{aligned}
\chi^0_{[\mathcal{Q},v]} &= |P_0 - 0| \vee |P_1 - 0|,\\
\chi^0_{[\mathcal{Q},w_0]} &= |P_0 - 0| \vee |P_1 - 2|,\\
\chi^0_{[\mathcal{Q},w_1]} &= |P_0 - 1| \vee |P_1 - 0|,\\
\chi^0_{[\mathcal{Q},w_2]} &= |P_0 - 2| \vee |P_1 - 3|,\\
\chi^0_{[\mathcal{Q},w_3]} &= |P_0 - 3| \vee |P_1 - 1|.
\end{aligned}$$

We only need formulae up to depth 1.

$$\begin{aligned}
\chi^1_{[\mathcal{Q},v]} =&\chi^0_{[\mathcal{Q},v]} \vee\\
&((\Box\chi^0_{[\mathcal{Q},w_0]}) \vee (\Box\chi^0_{[\mathcal{Q},w_1]}) \vee (\Box\chi^0_{[\mathcal{Q},w_2]}) \vee (\Box\chi^0_{[\mathcal{Q},w_3]}))\vee\\
&(\Diamond(\chi^0_{[\mathcal{Q},w_0]} \wedge \chi^0_{[\mathcal{Q},w_1]} \wedge \chi^0_{[\mathcal{Q},w_2]} \wedge \chi^0_{[\mathcal{Q},w_3]})).
\end{aligned}$$

If we evaluate the formula on $(\mathcal{Q}', v')$, we get 1 which is the bisimulation distance between the systems.

$$\begin{aligned}
[\![\chi^1_{[\mathcal{Q},v]}]\!]^{\mathcal{Q}'}(v') =&\max\{0, \max\{\min\{1,1,2,3\}, \min\{1,3,1,2\}, \min\{2,1,3,1\},\\
&\min\{3,2,1,1\}\}, \max\{\min\{1,1,2,3\}, \min\{1,3,1,2\},\\
&\min\{2,1,3,1\}, \min\{3,2,1,1\}\}\}\\
=&\max\{0, \max\{1,1,1,1\}, \max\{1,1,1,1\}\} = 1
\end{aligned}$$

Proposition 2.20. *For a given pointed finitely-branching quantitative transition system $(\mathcal{Q}, v)$ with only finitely many predicates and $\chi^n_{[\mathcal{Q},v]}$ constructed as described above, we have for every pointed quantitative transition system $(\mathcal{Q}', v')$ that whenever $[\![\chi^n_{[\mathcal{Q},v]}]\!]^{\mathcal{Q}'}(v') \leq r$ then $\mathrm{val}\mathcal{G}_{\sim_n}(\mathcal{Q}, v, \mathcal{Q}', v') \leq r$, i.e. Duplicator has a strategy in the n-round bisimulation game to guarantee an outcome at most r.*

Proof. We prove this proposition by induction on n, the number of rounds in the game. In the case that $n = 0$, we just have to consider the propositional distance of the two nodes where the pebbles are placed in the game and check that it is at most r. Since

$$[\![\chi^0_{[\mathcal{Q},v]}]\!]^{\mathcal{Q}'}(v') = [\![\bigvee_{i \in I} |P_i - P_i(v)|]\!]^{\mathcal{Q}'}(v') = \max_{i \in I} |P_i(v') - P_i(v)| \leq r$$

by definition, and this is exactly the definition of $\mathrm{pd}(v, v')$, the base case holds.

In the $n + 1$-st round of the game, we have to provide a strategy for Duplicator to guarantee the desired outcome (by using the strategy for the n-round game). We have that $[\![\chi^{n+1}_{[\mathcal{Q},v]}]\!]^{\mathcal{Q}'}(v') \leq r$. By definition, this means that

$$[\![\chi^{n+1}_{[\mathcal{Q},v]}]\!]^{\mathcal{Q}'}(v') = [\![\chi^0_{[\mathcal{Q},v]} \vee \bigvee_{w \in vE} \Box\chi^n_{[\mathcal{Q},w]} \vee \Diamond \bigwedge_{w \in vE} \chi^n_{[\mathcal{Q},w]}]\!]^{\mathcal{Q}'}(v')$$

$$= \max\{\mathrm{pd}(v, v'), \max_{w \in vE} \min_{w' \in v'E'} [\![\chi^n_{[\mathcal{Q},w]}]\!]^{\mathcal{Q}'}(v'), \max_{w' \in v'E'} \min_{w \in vE} [\![\chi^n_{[\mathcal{Q},w]}]\!]^{\mathcal{Q}'}(v')\} \leq r.$$

The first part $[\![\chi^0_{[\mathcal{Q},v]}]\!]^{\mathcal{Q}'}(v') = \mathrm{pd}(v, v') \leq r$ ensures that the propositional distance at current state is not exceeding r, so Duplicator has not already lost. Now she has to provide a move in her strategy against all possible moves Spoiler can make so that the outcome of the n-round game played from the next state is still less than or equal to r.

If Spoiler moves in $\mathcal{Q}'$, she will use that fact that the formula provides

$$[\![\Diamond \bigwedge_{w \in vE} \chi^n_{[\mathcal{Q},w]}]\!]^{\mathcal{Q}'}(v') = \max_{w' \in v'E'} \min_{w \in vE} [\![\chi^n_{[\mathcal{Q},w]}]\!]^{\mathcal{Q}'}(w'),$$

i.e. for every move (v', w') Spoiler makes in $\mathcal{Q}'$, she has a response (v, w) in $\mathcal{Q}$, such that $[\![\chi^n_{[\mathcal{Q},w]}]\!]^{\mathcal{Q}'}(w') \leq r$. This means by induction hypothesis that she has a strategy from w to guarantee an outcome for the n-round game of less than or equal to r.

If Spoiler moves in $\mathcal{Q}$, she will use that fact that the formula provides

$$[\![\bigvee_{w \in vE} \Box \chi^n_{[\mathcal{Q},w]}]\!]^{\mathcal{Q}'}(v') = \max_{w \in vE} \min_{w' \in v'E'} [\![\chi^n_{[\mathcal{Q},w]}]\!]^{\mathcal{Q}'}(w'),$$

i.e. for every move (v, w) Spoiler makes in $\mathcal{Q}$, she has a response (v', w') in $\mathcal{Q}'$, such that $[\![\chi^n_{[\mathcal{Q},w]}]\!]^{\mathcal{Q}'}(w') \leq r$ which again by induction hypothesis gives the desired strategy for the n-round game. $\Box$

As characteristic formulae of depth n are obviously included in QML^n and also clearly $[\![\chi^n_{[\mathcal{Q},v]}]\!]^{\mathcal{Q}}(v) = 0$ we can conclude the following.

Corollary 2.21. *For two finitely-branching pointed QTS with finitely many predicates $(\mathcal{Q}, v,)$ and $(\mathcal{Q}', v')$, if for all $\varphi \in$ QML^n, $|[\![\varphi]\!]^{\mathcal{Q}}(v) - [\![\varphi]\!]^{\mathcal{Q}'}(v')| \leq r$ then* val$\mathcal{G}_{\sim_n}(\mathcal{Q}, v, \mathcal{Q}', v') \leq r$. *Or in other words,*

$$(\mathcal{Q}, v) \equiv^r_{\mathrm{QML}^n} (\mathcal{Q}', v') \text{ implies } (\mathcal{Q}, v) \sim^r_n (\mathcal{Q}', v').$$

Altogether we can state the following (as in the classical case).

Theorem 2.22. *Let $(\mathcal{Q}, v)$ and $(\mathcal{Q}', v')$ be finitely-branching pointed quantitative transition systems with finitely many predicates and $r \in \mathbb{R}$. Then the following are equivalent:*

1. $(\mathcal{Q}, v) \sim^r_n (\mathcal{Q}', v')$
2. $[\![\chi^n_{[\mathcal{Q},v]}]\!]^{\mathcal{Q}'}(v') \leq r$
3. $(\mathcal{Q}, v) \equiv^r_{\mathrm{QML}^n} (\mathcal{Q}', v')$.

As a corollary, we obtain that finite r-bisimulation coincides with QML equivalence where finite bisimulation is defined as follows.

Definition 2.23. We call two pointed quantitative transition systems $(\mathcal{Q}, v)$ and $(\mathcal{Q}', v')$ *finitely r-bisimilar*, denoted $\sim^r_\omega$, if $(\mathcal{Q}, v) \sim^r_n (\mathcal{Q}', v')$ for all $n \in \mathbb{N}$.

Corollary 2.24. *Let $(\mathcal{Q}, v)$ and $(\mathcal{Q}', v')$ be finitely-branching pointed quantitative transition systems with finitely many predicates and $r \in \mathbb{R}$, then*

$$(\mathcal{Q}, v) \sim^r_\omega (\mathcal{Q}', v') \text{ if and only if } (\mathcal{Q}, v) \equiv^r_{\mathrm{QML}} (\mathcal{Q}', v').$$

Now we can prove a quantitative version of the Hennessy-Milner theorem [25], namely that for finitely-branching structures finite bisimulation implies full bisimulation equivalence. Hence, quantitative bisimulation coincides with quantitative modal equivalence (up to some real number *r*) for finitely-branching structures.

Theorem 2.25. *Let $(\mathcal{Q}, v)$ and $(\mathcal{Q}', v')$ be finitely-branching pointed quantitative transition systems with finitely many predicates and $r \in \mathbb{R}$. Then,*

$$(\mathcal{Q}, v) \sim^r_\omega (\mathcal{Q}', v') \text{ implies } (\mathcal{Q}, v) \sim^r (\mathcal{Q}', v').$$

Proof. As in the classical case this proof follows a game-based approach. Towards a contradiction assume that Spoiler has a strategy γ to get an outcome of $g > r$. Now, we consider the tree of all plays played consistent with γ. Each branch represents a play and in each of these plays, by our assumption, there comes a position $(v_j, {v_j}')$ with $\mathrm{pd}(v_j, {v_j}') > r$. Now, consider the tree of all play prefixes pruned at the positions where the propositional differences greater than *r* occur. As, by our assumption, the systems are finitely-branching, this tree is finite and thus has a finite height *n*. Thus, $(\mathcal{Q}, v) \not\sim^r_n (\mathcal{Q}', v')$ which is a contradiction to $(\mathcal{Q}, v) \sim^r_\omega (\mathcal{Q}', v')$.

□

3 Quantitative Parity Games and Model Checking

In this chapter, we introduce quantitative parity games, a generalisation of classical parity games. Classical parity games are two-player infinite games on coloured graphs. Finite plays are lost by the player who got stuck while infinite plays are decided by the colours that are seen infinitely often during the play. The main difference in the quantitative setting is that the games are no longer win-or-lose, but instead each play has a quantitative outcome. It is well-known, that classical parity games are model-checking games for the modal μ-calculus [15, 24], and our goal in this chapter is to extend this connection to the quantitative setting.

First, we introduce a suitable game model, then we show how to define model-checking games for a given formula and system within this setting. Classical parity games are determined, i.e. we can decide for each position which player wins from this position, there are no undecided positions. They are even positionally determined which means that the players do not need memory for their winning strategies [14, 35, 43]. In the quantitative setting, determinacy translates to the existence of the value of a game. It is not obvious that quantitative parity games are determined. We adapt a technique called unfolding which is used in the classical setting to prove the correctness of the model-checking games for L_μ and least fixed-point logic LFP [24]. In the quantitative setting, we use the unfolding technique to show the determinacy of quantitative parity games by providing strategies for both players and the correctness of our model-checking games. Unfortunately, quantitative parity games do not admit optimal strategies, but we show that for a fixed ε they enjoy ε-optimal positional strategies, i.e. memoryless strategies that allow the players to get an outcome close to the value. In the last section, we show a nice application of model-checking games by giving an alternative proof of the bisimulation invariance theorem from the previous chapter.

3.1 Quantitative Parity Games

Quantitative parity games extend classical parity games by enriching the game structure with a payoff function for terminal nodes, i.e. nodes without any outgoing edges. This gives real-valued payoffs for all finite plays. The payoff of an infinite play still only depends on the lowest priority seen infinitely often, as for classical parity games.

Definition 3.1. A *quantitative parity game* (QPG) is a tuple

$$\mathcal{G} = (V, V_0, V_1, E, \lambda, \Omega),$$

where the directed graph (V, E) is called the *game arena*. V is a disjoint union of V_0 and V_1, i.e. positions belong to either Player 0 or 1. The transition relation $E \subseteq V \times V$ describes possible moves in the game. The payoff function $\lambda : \{v \in V : vE = \emptyset\} \to \mathbb{R}_\infty$ assigns values to all terminal positions and the priority function $\Omega : V \to \{0, \dots, n\}$ assigns a priority to every position.

How to play. Every play starts at some position $v \in V$. For every position in V_i, Player i chooses a successor position, and the play proceeds from that position. If the play reaches a terminal position, it ends. We denote by $\pi = v_0, v_1, \dots$ the (possibly infinite) play through vertices $v_0, v_1, \dots$, given that $(v_n, v_{n+1}) \in E$ for every n. The outcome $\mathrm{p}(\pi)$ of a finite play $\pi = v_0 \dots v_k$ depends only on the value assigned to the terminal node by the payoff function λ, i.e. $\mathrm{p}(v_0, v_1, \dots, v_k) = \lambda(v_k)$.

The outcome $\mathrm{p}(\pi)$ of an infinite play depends only on the lowest priority seen infinitely often. We 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. To avoid confusion when using pronouns, we refer to Player 0 as "she" and Player 1 as "he".

Strategies. A *strategy* for Player $i \in \{0,1\}$ is a function $s : V^*V_i \to V$ with $(v, s(v_0, \dots, v_n, v)) \in E$ for $v \in V$ and a play prefix $v_0, \dots, v_n$. A play $\pi = v_0, v_1, \dots$ is *consistent with a strategy* s for player i, if $v_{n+1} = s(v_0, \dots, v_n)$ for every n such that $v_n \in V_i$. For strategies σ, ρ of the two players, we denote by $\pi_{\sigma,\rho}(v)$ the unique play starting at node v which is consistent with both σ and ρ.

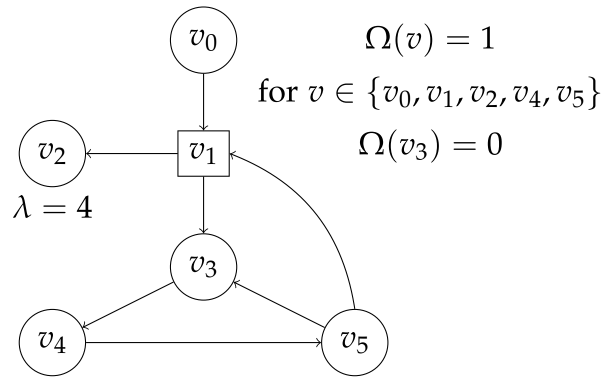


Figure 3.1: A quantitative parity game $\mathcal{G} = (V, V_0, V_1, E, \lambda, \Omega)$

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

$$\sup_{\sigma \in \Gamma_0} \inf_{\rho \in \Gamma_1} \mathrm{p}(\pi_{\sigma,\rho}(v)) = \inf_{\rho \in \Gamma_1} \sup_{\sigma \in \Gamma_0} \mathrm{p}(\pi_{\sigma,\rho}(v)) =: \mathrm{val}\mathcal{G}(v),$$

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* v, $\mathrm{val}\mathcal{G}(v)$.

For a determined game $\mathcal{G}$, we call a strategy σ of Player 0 *optimal* if for every strategy ρ of Player 1 $\mathrm{p}(\pi_{\sigma,\rho}(v)) \geq \mathrm{val}\mathcal{G}(v)$ (and analogously for Player 1).

Example 3.2. In Figure 3.1, we show a simple quantitative parity game. By convention, we depict positions of Player 0 by circles and positions of Player 1 by rectangles. Let us look at a finite play starting at v_0. This position belongs to Player 0 and she has to move to v_1. This position belongs to Player 1, if he decides to move to v_2 the play ends as this is a terminal position, thus resulting in $\pi = v_0, v_1, v_2$ with payoff $\mathrm{p}(\pi) = \lambda(v_2) = 4$.

Player 1 could at v_1 also decide to go to v_3. Then, Player 0 can decide to stay in the cycle v_3, v_4, v_5 forever, and not give Player 1 another opportunity to end the play by moving back to v_1. The payoff of this infinite play $\pi = v_0, v_1, (v_3, v_4, v_5)^\omega$ is $\mathrm{p}(\pi) = \infty$ as the smallest priority occurring infinitely often is 0.

This shows that it is an optimal strategy for Player 1 to end the play by moving to v_2 directly (and not giving Player 0 any chance to force an infinite play), as the value of the game starting at v_0 is 4.

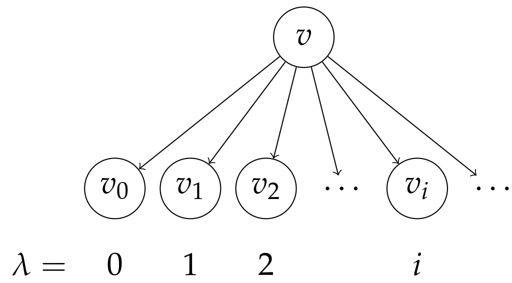


Figure 3.2: An infinitely-branching QPG where Player 0 has no optimal strategy

Classical parity games can be seen as a special case of quantitative parity games on qualitative arenas. As in qualitative transition systems, we only have payoffs ∞ and $-\infty$ in terminal nodes. Then, we map winning to payoff ∞ and losing to payoff $-\infty$. Classical determinacy is recovered by observing that only two values are possible and that guaranteeing one of those exactly means that one of the players has a winning strategy from every position of the game, i.e., the game arena is partitioned into winning regions for both players.

Formally, we say that a quantitative parity game $\mathcal{G} = (V, V_0, V_1, E, \lambda, \Omega)$ is *qualitative* if $\lambda(v) = -\infty$ or $\lambda(v) = \infty$ for all $v \in V$ with $vE = \emptyset$. In qualitative games, we denote by $W_i \subseteq V$ the winning region of Player i, i.e. W_0 is the region where Player 0 has a strategy to guarantee payoff ∞ and W_1 is the region where Player 1 can guarantee payoff $-\infty$.

Qualitative parity games have been extensively studied in the past. One of their fundamental properties is *positional determinacy* [14, 35, 43]. In every parity game, the set of positions can be partitioned into the winning regions W_0 and W_1 for the two players, and each player has a positional winning strategy on her winning region (which means that the moves selected by the strategy only depend on the current position, not on the history of the play).

Winning strategies translate into optimal strategies in the quantitative world. Unfortunately, the existence of optimal strategies is not guaranteed in quantitative parity games – unless we restrict the payoff functions to finite domains. Below, we give two examples of infinite games where Player 0 has no optimal strategy. However, there exist ε-optimal strategies, a concept that we introduce later in Section 3.3.

Example 3.3. In Figure 3.2, we show a very simple infinitely-branching QPG. In

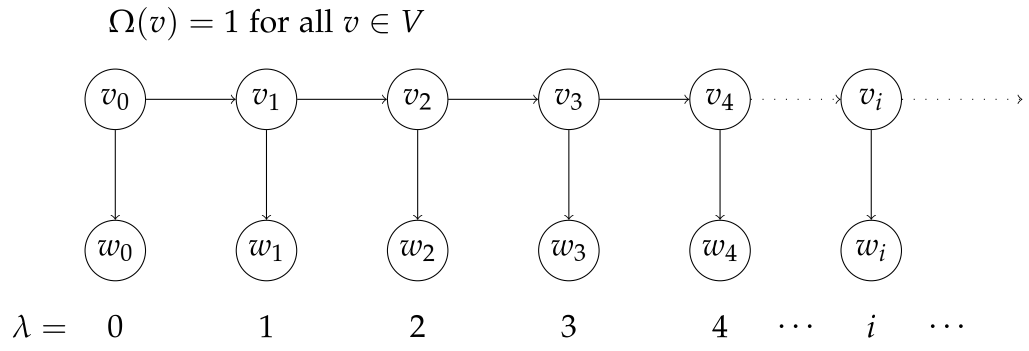


Figure 3.3: A finitely-branching QPG where Player 0 has no optimal strategy

this game, only Player 0 moves. She has exactly one choice to make, namely to move from v to one of the infinitely-many successors. The value of the game is the supremum over all the payoffs she could achieve by using different strategies. As she can achieve arbitrarily high payoffs, this value is ∞. But there is no strategy that will give Player 0 this payoff (as no position actually has payoff ∞), so she has to settle for a payoff $i \in \mathbb{N}$. She still has an ε-optimal strategy for every $\varepsilon \in (0, 1)$ (as defined in Definition 3.9), i.e. strategies that give her a payoff greater or equal to $\frac{1}{\varepsilon}$.

Example 3.4. In Figure 3.3, we illustrate that also finitely-branching (infinite) games do not necessarily admit optimal strategies. In this game, again only Player 0 has any choice. In every step, she can decide if she is happy with the value she will get if she ends the play by moving down to a terminal position, or if she wants to continue playing. She cannot, however, move along the infinite path forever, as the only priority in this game is odd so an infinite play would be the worst case for her. It can again easily be seen that the value of the game is ∞, but that there is no play with this payoff, so she has no optimal strategy. She has, however, as before, strategies to get arbitrarily high payoffs.

Please note that if we allow quantities on edges as well, as we will do in Chapter 4, the situation gets worse as even very simple finite games do not admit optimal strategies anymore.

Although we do not have positional optimal strategies, we can show that for a fixed ε, quantitative parity games admit positional ε-optimal strategies. As we need some more concepts and definitions to prove this, we come back to this later in Theorem 3.29.

3.2 Model-Checking Games for Qμ

A game $(\mathcal{G}, v)$ is a model-checking game for a formula φ and a structure $\mathcal{Q}, v'$, if the value of the game starting from v is exactly the value of the formula evaluated on $\mathcal{Q}$ at v'. In the qualitative case, that means that φ holds in $\mathcal{Q}, v'$ if Player 0 wins in $\mathcal{G}$ from v.

Model-checking games give us a more intuitive way to look at the evaluation of Qμ formulae. Few will deny that Qμ and L$_\mu$ formulae are very un-intuitive and hard to understand as soon as they involve more than one quantifier alternation. To construct a game out of a given formula and system often makes it much easier to understand the meaning of the formula.

But this is only one nice feature and not the main advantage model-checking games give us. The game setting also allows us to use a whole new tool box, namely to adopt game-theoretic techniques and methods to solve the model-checking problem. This also paves the way for an algorithmic treatment.

Definition 3.5. For a quantitative transition system $\mathcal{Q} = (S, T, P_i)$ and a closed Qμ-formula φ in negation normal form, the quantitative parity game

$$\mathrm{MC}[\mathcal{Q}, \varphi] = (V, V_0, V_1, E, \lambda, \Omega),$$

which we call the *model-checking game* for $\mathcal{Q}$ and φ, is constructed in the following way.

Positions. The positions of the game are the pairs (ψ, s), where ψ is a subformula of φ, and $s \in S$ is a state of the QTS $\mathcal{Q}$, and the two special positions $(-\infty)$ and (∞). Positions (ψ, s) where the top operator of ψ is $\Box$, $\wedge$, or ν belong to Player 1 and all other positions belong to Player 0.

Moves. Positions of the form $(P_i + c, s)$, $(\neg(P_i + c), s)$, $(-\infty)$, and (∞) are terminal positions. From positions of the form $(\psi \wedge \vartheta, s)$, resp. $(\psi \vee \vartheta, s)$, one can move to (ψ, s) or to (ϑ, s). Positions of the form $(\Diamond\psi, s)$ have either a single successor $-\infty$, in case s is a terminal state in $\mathcal{Q}$, or one successor (ψ, s') for every $s' \in sT$. Analogously, positions of the form $(\Box\psi, s)$ have a single successor (∞), if $sT = \emptyset$, or one successor (ψ, s') for every $s' \in sT$ otherwise. Fixed-point positions $(\mu X.\psi, s)$ or $(\nu X.\psi, s)$ have a single successor (ψ, s). Whenever one encounters a position where the fixed-point variable stands alone, i.e. (X, s'), the play goes back to the corresponding definition, namely (ψ, s').

Payoffs. The payoff function λ assigns $[\![P_i]\!](s) + c$ to positions $(P_i + c, s)$, $-([\![P_i]\!](s) + c)$ to positions $(\neg(P_i + c), s)$, ∞ to position (∞), and $-\infty$ to position $(-\infty)$.

Priorities. The priority function Ω is defined as in the classical case using the alternation level of the fixed-point variables, as described in [24]. Positions (X, s) get a lower priority than positions (X', s') if X has a lower alternation level than X'. Ω assigns an even value to all positions (X, s), where X is a ν-variable and an odd value to all positions (X, s), where X is a μ-variable, using the alternation level of the fixed-point variables as priorities. To adjust the alternation level to get the right parity we set $\mathrm{al}^*_\varphi(X) = \mathrm{al}_\varphi(X) + 1$ if μ-variables are at even alternation levels (or equivalently ν-variables at odd alternation levels) and $\mathrm{al}^*_\varphi(X) = \mathrm{al}_\varphi(X)$ otherwise. Let $\mathrm{ad}^*_\varphi = \max\{\mathrm{al}^*(X) : X \text{ is a fixed-point variable in } \varphi\}$ and set $\Omega(X, s) := \mathrm{al}^*_\varphi(X)$ for positions, where X is a fixed-point variable and $\Omega(p) := \mathrm{ad}^*_\varphi$ for all other positions.

It is well-known that qualitative parity games are model-checking games for the classical μ-calculus [15]. A proof that uses the unfolding technique can be found in [24].

Theorem 3.6. *For a formula φ in L_μ, a qualitative transition system $\mathcal{K}$ and a node v, $\mathrm{MC}[\mathcal{K}, \varphi]$ starting from (φ, v) is a model-checking game for φ and $\mathcal{K}, v$, i.e., Player 0 (or Verifier) has a winning strategy in $\mathrm{MC}[\mathcal{K}, \varphi]$ from (φ, v) if and only if $\mathcal{K}, v \models \varphi$.*

We generalise this connection to the quantitative setting as follows.

Theorem 3.7. *For a formula φ in $\mathrm{Q}\mu$, a quantitative transition system $\mathcal{Q}$, and $v \in \mathcal{Q}$, the game $\mathrm{MC}[\mathcal{Q}, \varphi]$ is determined and*

$$\mathrm{valMC}[\mathcal{Q}, \varphi](\varphi, v) = [\![\varphi]\!]^{\mathcal{Q}}(v).$$

By using the method of unfolding, we give a direct proof for both the determinacy of quantitative parity games and the connection to model checking $\mathrm{Q}\mu$. An alternative method to prove the determinacy of quantitative parity games is via the determinacy of classical parity games. To apply this result here, consider for a quantitative game $\mathcal{G}$ the infinite family of two-valued, i.e. zero-sum games $\mathcal{G}_r$ for $r \in \mathbb{R}_\infty$. Each $\mathcal{G}_r$ is identical to $\mathcal{G}$, except that Player 0 wins a play in $\mathcal{G}_r$ if the payoff of the corresponding play in $\mathcal{G}$ is at least r, and Player 1 wins in the other case. A formal definition can be found in the proof of Theorem 3.29. As classical parity games are determined, it follows that each game $\mathcal{G}_r$ is

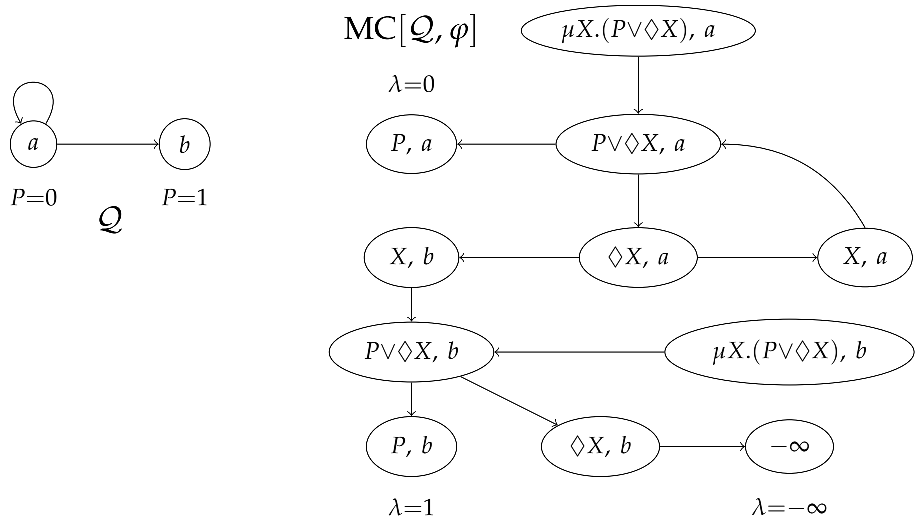


Figure 3.4: Model-checking game for $\varphi = \mu X.(P \vee \Diamond X)$ and $\mathcal{Q}$.

determined. The value of $\mathcal{G}$ can be calculated as the supremum of all r such that Player 0 wins $\mathcal{G}_r$ and thus the determinacy of $\mathcal{G}$ follows. Please observe that it is impossible that Player 1 wins $\mathcal{G}_a$ and Player 0 wins $\mathcal{G}_b$ for $b > a$, as a winning strategy for Player 0 for $\mathcal{G}_b$ would also be winning in $\mathcal{G}_a$.

Next, we give an example to illustrate the construction of a model-checking game.

Example 3.8. A model-checking game for $\varphi = \mu X.(P \vee \Diamond X)$ on the QTS $\mathcal{Q}$ is shown in Figure 3.4. The nodes are labelled with the corresponding subformulae of φ, and the state of $\mathcal{Q}$. As there is only one fixed-point variable and no alternation, we only need one priority. Since it is referring to a least fixed-point, the (adjusted) priority will be 1 and assigned not only to the fixed-point position but also to all other nodes. This means that infinite plays are bad for player Player 0. As we know that this formula classically describes a reachability property (P is reachable in a finite number of steps) this captures our intuition that Player 0 should be punished if she fails to establish a value in a finite number of steps. Note that in this game Player 0 is the only one allowed to make any choices. When we start at the top node, corresponding to an evaluation of φ at a in $\mathcal{Q}$, the only choice she has is either to keep playing (by looping), or to end the game by moving to a terminal position. The only reasonable strategy is to move directly to the position (P, b) which will give her a payoff of 1. This is equal to $[\![\varphi]\!]^{\mathcal{Q}}(a)$.

3.3 Unfolding Quantitative Parity Games

We prove the determinacy of quantitative parity games by an induction on the number of priorities occurring in the game. The base case are games with one priority. We call these games *reachability games* if the only priority is odd and *safety games* if it is even. As the games are dual to each other, we focus on reachability games and show that they can be solved by a generalisation of backwards induction.

For the induction step, we adapt the classical method of unfolding a parity game. This technique allows us to transform a game with m priorities into a sequence of games with $m-1$ priorities. Then, we show that the value of the original game can be computed from the values of the games of the unfolding.

After having established the determinacy of quantitative parity games, we prove that they correctly describe the evaluation of Qμ formulae. First, we look at formulae without fixed-point operators and thus give a proof for quantitative modal logic. For the fixed-point case, we show that the values of the games of the unfolding are closely related to the stages of the fixed-point evaluation of Qμ.

Let us first fix some notation and show a few basic properties. We need the notion of ε-optimal strategies. These are strategies that guarantee an outcome ε-close to the value of the game. As the value can be infinite in our setting we use the following notion of "close".

Definition 3.9. A number $k \in \mathbb{R}_\infty$ is called *ε-close* to $p \in \mathbb{R}_\infty$, when either p is finite and $|k-p| \leq \varepsilon$, $p = \infty$ and $k \geq \frac{1}{\varepsilon}$ or $p = -\infty$ and $k \leq -\frac{1}{\varepsilon}$. Furthermore, we say that k is *ε-above* p (or *ε-below*), if $k \geq p'$ (or $k \leq p'$) for some p' that is ε-close to p.

We slightly abuse the word "close" as ε-closeness is *not* symmetric, since $\frac{1}{\varepsilon}$ is ε-close to ∞, but ∞ is not ε-close to any number $r \in \mathbb{R}$ (and analogously for $-\infty$). Intuitively, if a value k is ε-above p it means that k is at least greater than a threshold p' ε-close to p (ε-below is defined analogously).

Definition 3.10. We call a strategy σ of Player 0 in a determined game $\mathcal{G}$ *ε-optimal* for $\varepsilon \in (0,1)$ if for every strategy ρ of Player 1, $\mathrm{p}(\pi_{\sigma,\rho}(v))$ is ε-close to $\mathrm{val}\mathcal{G}(v)$ (and analogously for Player 1).

In our proofs, we combine ε-optimal strategies and have to guarantee that the resulting strategy is still a good enough approximation of the original value. For this purpose, we note that closeness is transitive in a weaker sense.

Observation 3.11. *Let $x, y, z \in \mathbb{R}_\infty$, $\varepsilon \in (0,1)$. If x is $\varepsilon/2$-close to y and y is $\varepsilon/2$-close to z, then x is ε-close to z.*

This statement remains valid if we replace the close-relation by the above- or below-relation.

Reachability Games

This is the base case of our induction. In quantitative reachability games the only priority is odd and is assigned to every node, i.e. infinite plays have outcome $-\infty$. To determine the value of a play in a reachability game after k steps, we use backwards induction, and inductively define a sequence of approximate payoff functions $f_k : V \to \mathbb{R}_\infty$. Then, we show how to construct strategies for the two players to achieve this payoff and thus prove determinacy.

The first payoff function corresponds to the immediate payoff of the game, i.e. after 0 steps.

$$f_0(v) = \begin{cases} \lambda(v) & \text{for } vE = \emptyset \\ -\infty & \text{otherwise} \end{cases}$$

The payoff function f_{k+1} corresponds to the payoff that can be guaranteed for Player 0 in $k+1$ steps.

$$f_{k+1}(v) = \begin{cases} \lambda(v) & \text{for } vE = \emptyset \\ \sup_{w \in vE} f_k(w) & \text{for } v \in V_0 \\ \inf_{w \in vE} f_k(w) & \text{for } v \in V_1 \end{cases}$$

Intuitively, the functions f_k determine the value of a game that ends after at most k steps. This means that the outcome is $-\infty$ if Player 0 does not succeed to reach a terminal position in at most k steps.

We define $f(v)$ as $\lim_{k\to\infty} f_k(v)$ and we show that $f(v)$ is indeed the value of a game started at v. To show that $f(v)$ is well-defined, note that the sequence f_k is monotonically increasing, i.e. $f_{k+1}(v) \geq f_k(v)$ for all v, k. This can easily be proved by induction on k. Moreover, by commutativity of lim and sup, inf, the properties in the definition of f_k are maintained in f.

Lemma 3.12. *For all moves $(v, w) \in E$,*

$$f(v) = \begin{cases} \lambda(v) & \text{for } vE = \emptyset \\ \sup_{w \in vE} f(w) & \text{for } v \in V_0 \\ \inf_{w \in vE} f(w) & \text{for } v \in V_1 \end{cases} .$$

Let us now introduce strategies for the two players to approximate f_k and f. For given ε and k, the strategy σ_ε^k for Player 0 approximates the payoff f_k whereas the strategy ρ_ε for Player 1 approximates f. For σ_ε^k choose at v a successor node $w \in vE$ such that $f_{k-1}(w)$ is $\frac{\varepsilon}{2}$-close to $f_k(v)$. From w on, play according to $\sigma_{\frac{\varepsilon}{2}}^{k-1}$. For ρ_ε in v proceed in an analogous way: choose a successor node w such that $f(w)$ is $\frac{\varepsilon}{2}$-close to $f(v)$ and play $\rho_{\frac{\varepsilon}{2}}$ from w. When the opponent makes a move, adjust ε to $\frac{\varepsilon}{2}$ in the same way.

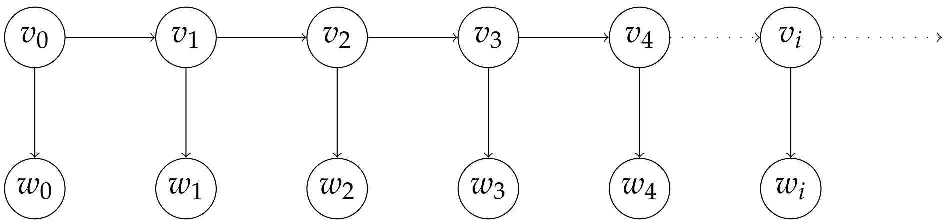
Note that if the game is finitely-branching, the ε-approximations are not necessary as one can choose the maximal and minimal value directly.

Before we proceed, let us illustrate the construction of f_k, f and σ_ε^k by an example.

Example 3.13. In Figure 3.5, we see again the infinite quantitative parity game from Example 3.4. We recall that Player 0 has no optimal strategy, but that she can achieve arbitrarily high values, thus she has an ε-optimal strategy for every ε. We want to illustrate the computation of the values $f_k(v)$ for all positions $v \in V$ and how to construct a strategy σ_ε^k for given k and ε. The values of the functions f_k are depicted in the figure. $f(v_i) = \lim_{k\to\infty} f_k(v_i) = \infty$ for all non-terminal nodes v_i and $f(w_i) = \lambda(w_i)$ for all terminal nodes w_i.

Now let us consider a strategy σ_ε^k for Player 0, e.g. for $k = 3$ and $\varepsilon = \frac{1}{2}$. This means that for a play starting at v_0 this strategy guarantees an outcome p which is $\frac{1}{2}$-above $f_3(v_0) = 2$, i.e. $p \geq 2 - \frac{1}{2} = 1.5$. The construction of $\sigma_{\frac{1}{2}}^3$ proceeds as follows. $\sigma_{\frac{1}{2}}^3(v_0)$ has to choose a $w \in v_0E$ such that $f_2(w)$ is $\frac{1}{4}$-close to $f_3(v_0) = 2$, i.e. $f_2(w) \geq 2 - \frac{1}{4}$. This leaves us with v_1 and from here we play according to $\sigma_{\frac{1}{4}}^2$. Now, $\sigma_{\frac{1}{4}}^2(v_1)$ has to choose a $w \in v_1E$ such that $f_1(w)$ is $\frac{1}{8}$-close to $f_2(v_1) = 2$. This gives us v_2 and we change to $\sigma_{\frac{1}{8}}^1$. Again, at v_2, we have to choose a $w \in v_2E$ such that $f_0(w)$ is $\frac{1}{16}$-close to $f_1(v_2) = 2$. As $f_0(v_3) = -\infty$, this forces us to choose the terminal node w_2 with $f_0(w_2) = \lambda(w_2) = 2$. This gives us the desired outcome as 2 is $\frac{1}{2}$-close to 2. Please note that this strategy is $\frac{1}{2}$-optimal in $\mathcal{G}$ starting from v_0 as $\mathrm{val}\mathcal{G}(v_0) = \infty$ and we have defined $\frac{1}{2}$-close to ∞ as greater or equal to 2. This also implies that for every ε, she has an

$\Omega(v) = 1$ for all $v \in V$



$\lambda =$ 0 1 2 3 4 $\cdots$ i $\cdots$

	v_0	w_0	v_1	w_1	v_2	w_2	v_3	$\ldots$	v_i	w_i	$\ldots$
f_0	$-\infty$	0	$-\infty$	1	$-\infty$	2	$-\infty$		$-\infty$	i	
f_1	0	0	1	1	2	2	3		i	i	
f_2	1	0	2	1	3	2	4		$i+1$	i	
f_3	2	0	3	1	4	2	5		$i+2$	i	
f_4	3	0	4	1	5	2	6		$i+3$	i	
$\vdots$									$\vdots$		
f	∞	0	∞	1	∞	2	∞	$\ldots$	∞	i	$\ldots$

Figure 3.5: Illustration of the strategy construction for Player 0

ε-optimal strategy, she just has to take more steps along the path of v_i before moving to a terminal position to get an outcome greater or equal to $\frac{1}{\varepsilon}$. Please note that we presented the general construction for the strategy σ_ε^k, although in this game those approximations are not necessary, as the game graph is finitely-branching.

Lemma 3.14. $\mathrm{p}(\pi_{\sigma_\varepsilon^k,\rho}(v))$ *is* ε*-above* $f_k(v)$ *for every strategy* ρ *of Player* 1.

Proof. We prove this lemma by induction on k. For $k = 0$, $f_0(v) \neq -\infty$ only in case that v is a terminal node, but then $f_0(v) = \lambda(v) = \mathrm{p}(\pi_{\sigma_\varepsilon^0,\rho}(v))$.

By induction hypothesis $\mathrm{p}(\pi_{\sigma_{\frac{\varepsilon}{2}}^{k-1},\rho}(w))$ is $\frac{\varepsilon}{2}$-above $f_{k-1}(w)$ for all $w \in V$ and $\varepsilon \in (0,1)$.

If $v \in V_0$, then σ_ε^k by definition chooses w such that $f_{k-1}(w)$ is $\frac{\varepsilon}{2}$-close to $f_k(v)$. Thus, by the above and Observation 3.11, $\mathrm{p}(\pi_{\sigma_\varepsilon^k,\rho}(v)) = \mathrm{p}(\pi_{\sigma_{\frac{\varepsilon}{2}}^{k-1},\rho}(w))$ is ε-above $f_k(v)$.

If $v \in V_1$, then ρ chooses any successor w', by definition

$$f_k(v) = \inf_{w \in vE} f_{k-1}(w) \leq f_{k-1}(w'),$$

and thus $\mathrm{p}(\pi_{\sigma_\varepsilon^k,\rho}(v)) = \mathrm{p}(\pi_{\sigma_{\frac{\varepsilon}{2}}^{k-1},\rho}(w'))$ is even $\frac{\varepsilon}{2}$-above $f_k(w)$. □

Lemma 3.15. *The strategies* $\sigma_{\frac{\varepsilon}{2}}^k$ *are* ε*-optimal, i.e. for every* ε *and* v *there is a* k *such that* $\mathrm{p}(\pi_{\sigma_{\frac{\varepsilon}{2}}^k,\rho}(v))$ *is* ε*-above* $f(v)$ *for every strategy* ρ *of Player* 1.

Proof. As $f(v) = \lim_{i\to\infty} f_i(v)$, for every ε there is a k such that $f_k(v)$ is $\frac{\varepsilon}{2}$-close to $f(v)$. By Lemma 3.14, $\mathrm{p}(\pi_{\sigma_{\frac{\varepsilon}{2}}^k,\rho}(v))$ is $\frac{\varepsilon}{2}$-above $f_k(v)$ and thus by Observation 3.11 it is ε-above $f(v)$. □

Lemma 3.16. *The strategy* ρ_ε *is* ε*-optimal, i.e.* $\mathrm{p}(\pi_{\sigma,\rho_\varepsilon}(v))$ *is* ε*-below* $f(v)$ *for every strategy* σ *of Player* 0.

Proof. Towards a contradiction, assume there is a σ such that

$$\mathrm{p}(\pi_{\sigma,\rho_\varepsilon}(v)) > f(v) + \varepsilon.$$

Then $\mathrm{p}(\pi_{\sigma,\rho}(v)) > -\infty$, and thus $\pi_{\sigma,\rho}(v)$ is finite.

We show by induction on the length k of $\pi_{\sigma,\rho}(v)$ that $\mathrm{p}(\pi_{\sigma,\rho_\varepsilon}(v))$ is ε-below $f(v)$. The case $k = 0$ means that v is a terminal position and then

$$\mathrm{p}(\pi_{\sigma,\rho_\varepsilon}(v)) = \lambda(v) = f(v).$$

For the induction step, let

$$\pi_{\sigma,\rho_\varepsilon}(v_0) = v_0, v_1, \dots, v_k.$$

By definition of ρ_ε, the play $v_1 \dots v_k$ is played consistent with $\rho_{\frac{\varepsilon}{2}}$ and therefore, by inductive assumption, $\mathrm{p}(v_1, \dots, v_k)$ is $\frac{\varepsilon}{2}$-below $f(v_1)$.

If $v_0 \in V_0$, then $f(v_0) = \sup_{w \in v_0 E} f(w) \geq f(v_1)$ and so $\mathrm{p}(v_0, v_1, \dots, v_k)$ is even $\frac{\varepsilon}{2}$-below $f(v_0)$.

If $v_0 \in V_1$, then, by definition, ρ_ε chooses a v_1 so that $f(v_1)$ is $\frac{\varepsilon}{2}$-close to $f(v_0)$ and thus, by Observation 3.11, $\mathrm{p}(v_0, v_1, \dots, v_k)$ is ε-below $f(v_0)$. □

As for all $v \in V$, both players have strategies that guarantee an outcome ε-close to $f(v)$, we can conclude that this is indeed the value of the game.

Proposition 3.17. *Reachability and safety games are determined, for every position v there exist strategies σ^ε and ρ^ε that guarantee payoffs ε-above (or respectively ε-below)* $\mathrm{val}\mathcal{G}(v)$.

Unfolding Step

We proceed with the induction step of our determinacy proof. To this end, we present a method to unfold a quantitative parity game with m priorities into a sequence of games with $m - 1$ priorities. The value of the original game can be computed from the values of the games in the unfolding. To prove this, we construct strategies for both players to achieve an approximation of this value. The unfolding technique is inspired by the proof of correctness of the model-checking games for L_μ and LFP in [24]. Besides establishing determinacy, we also use this method to prove the fixed-point case of Theorem 3.7. We show that, as in the classical case, the unfolding of $\mathrm{MC}[\mathcal{Q}, \varphi]$ is closely related to the inductive evaluation of fixed points in φ on $\mathcal{Q}$.

From now on, we assume that the minimal priority in $\mathcal{G}$ is even and call it m. This is no restriction, since, if the minimal priority is odd, we can always consider the dual game, where the roles of the players are switched and all priorities are decreased by one.

Definition 3.18. For a quantitative parity game $\mathcal{G} = (V, E, \lambda, \Omega)$, we define the *truncated game*,

$$\mathcal{G}^- = (V, E^-, \lambda, \Omega^-).$$

We assume without loss of generality that all nodes with minimal priority in $\mathcal{G}$ have unique successors. In $\mathcal{G}^-$ we remove the outgoing edge from each of these nodes. Since these nodes are terminal positions in $\mathcal{G}^-$, their priority does not matter any more for the outcome of a play and Ω^- assigns them a higher priority, e.g. $m+1$. Formally,

$$E^- = E \setminus \{(v, v') : \Omega(v) = m\}$$
$$\Omega^-(v) = \begin{cases} \Omega(v) & \text{if } \Omega(v) \neq m, \\ m+1 & \text{if } \Omega(v) = m. \end{cases}$$

The *unfolding of* $\mathcal{G}$ is a sequence of games $\mathcal{G}^-_\alpha$, for ordinals α, which all coincide with $\mathcal{G}^-$, except for the payoff functions λ_α. Below we give the construction of the λ_α.

For all terminal nodes v of the original game $\mathcal{G}$ we have $\lambda_\alpha(v) = \lambda(v)$ for all α. For the new terminal nodes, i.e. all $v \in V$ such that $vE^- = \emptyset$ and $vE = \{w\}$, the valuation is given by:

$$\lambda_\alpha(v) = \begin{cases} \infty & \text{for } \alpha = 0, \\ \mathrm{val}\mathcal{G}^-_{\alpha-1}(w) & \text{for } \alpha \text{ successor ordinal}, \\ \lim_{\beta<\alpha} \mathrm{val}\mathcal{G}^-_\beta(w) & \text{for } \alpha \text{ limit ordinal}. \end{cases}$$

The intuition behind the definition of λ_α is to give an incentive for Player 0 to reach the new terminal nodes by first giving them the best possible valuation, and later by updating them to values of their successor in a previous game $\mathcal{G}^-_\beta, \beta < \alpha$.

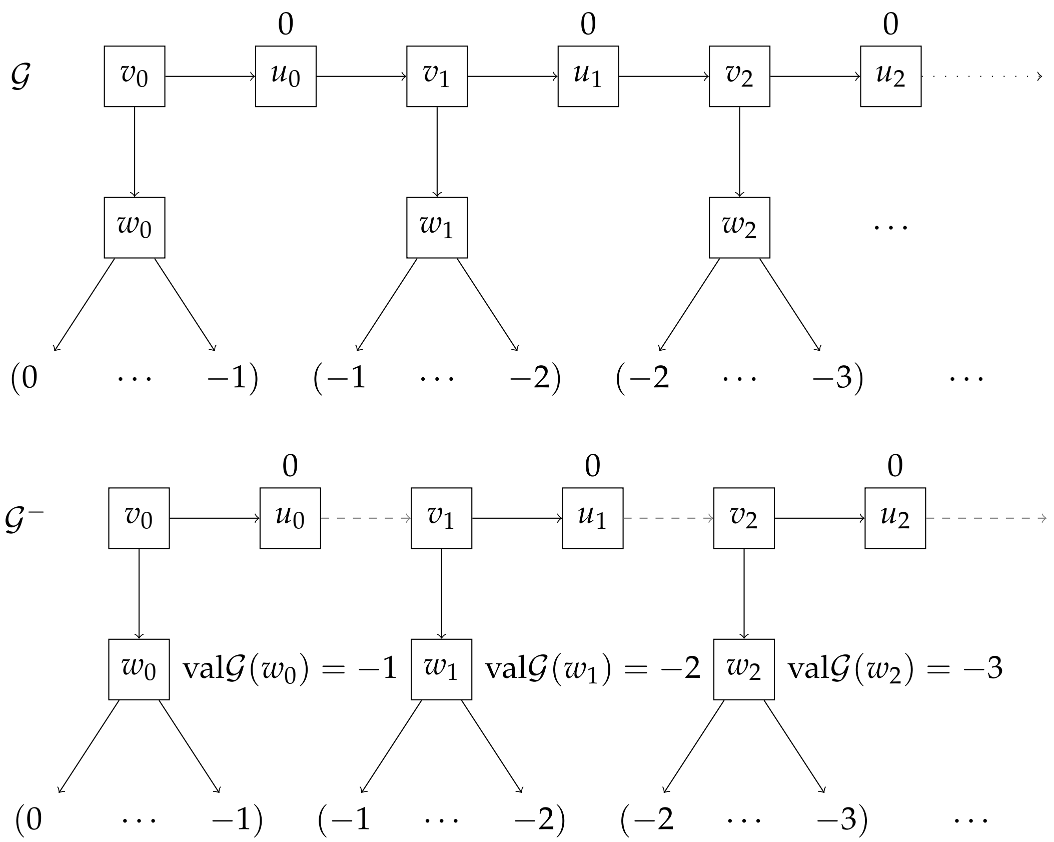
To determine the value of the original game $\mathcal{G}$, we inductively compute the values for each game in $\mathcal{G}_\alpha$, until they do not change any more.

Let γ be an ordinal for which $\mathrm{val}\mathcal{G}^-_\gamma = \mathrm{val}\mathcal{G}^-_{\gamma+1}$. Such an ordinal exists, since the values of the games in the unfolding are monotonically decreasing (which follows from determinacy of these games and the definition). We set

$$g(v) = g_\gamma(v) = \mathrm{val}\mathcal{G}^-_\gamma(v)$$

and show that g is the value function of the original game $\mathcal{G}$.

Example 3.19. In Figure 3.6, we want to illustrate how to unfold a quantitative parity game and how to compute the value of the original game via the games of the unfolding. We depict an infinite quantitative parity game $\mathcal{G} = (V, V_0, V_1, E, \lambda, \Omega)$ where all positions belong to Player 1, i.e. $V_0 = \emptyset$



$$
\begin{array}{lllll}
\mathrm{val}\mathcal{G}_0^-(v_0) = -1 & \lambda_0(u_0) = \infty & \mathrm{val}\mathcal{G}_0^-(v_1) = -2 & \lambda_0(u_1) = \infty & \mathrm{val}\mathcal{G}_0^-(v_2) = -3 \\
\mathrm{val}\mathcal{G}_1^-(v_0) = -2 & \lambda_1(u_0) = -2 & \mathrm{val}\mathcal{G}_1^-(v_1) = -3 & \lambda_1(u_1) = -3 & \mathrm{val}\mathcal{G}_1^-(v_2) = -4 \\
\mathrm{val}\mathcal{G}_2^-(v_0) = -3 & \lambda_2(u_0) = -3 & \mathrm{val}\mathcal{G}_2^-(v_1) = -4 & \lambda_2(u_1) = -4 & \mathrm{val}\mathcal{G}_2^-(v_2) = -5 \\
 & \vdots & & \vdots &
\end{array}
$$

$$\mathrm{val}\mathcal{G}_\infty^-(v_i) = -\infty = g(v_i) = \mathrm{val}\mathcal{G}_\infty^-(u_i) = g(u_i) \text{ for all } i \in \mathbb{N}$$

Figure 3.6: Unfolding for a QPG $\mathcal{G}$

and $V_1 = V$. Each w_i has infinitely many successors from the open interval $(-i, -(i+1))$, e.g. w_1 has successors for each $r \in (-1, -2)$ and each of these successors has payoff r. $\Omega : V \to \{0, 1\}$ assigns 0 to all nodes u_i and 1 to all other nodes. Hence, the nodes u_i are the nodes with minimal priority, so the unique outgoing edge from each of these nodes is removed in the truncated game $\mathcal{G}^-$. In the figure, we also show the values of the games in the unfolding.

Let us look at v_0. In the first game, $\mathcal{G}_0^-$, we have $\mathrm{val}\mathcal{G}_0^-(v_0) = -1$ as the only choice he has is to either go to u_0 or to w_0. Since $\lambda_0(u_0) = \infty$, the only reasonable choice is to go to w_0. The value of this position is -1, but there is no successor with this payoff. Please recall that $\lambda_0(u_i) = \infty$ for all nodes of even minimal priority u_i by definition to encourage Player 0 to seek out these positions and to discourage Player 1 from moving to them. In the next game, $\mathcal{G}_1^-$, the payoff function λ_1 is updated to $\lambda_1(u_0) = -2$, the value at v_1 in $\mathcal{G}_0^-$. Now it is a better choice for Player 1 to move to u_0 and thus $\mathrm{val}\mathcal{G}_1^-(v_0) = -2$. In the next step the payoff function is again updated to a smaller value, $\lambda_2(u_0) = -3$. Thus, the values that Player 1 can achieve from v_0 decrease with every iteration. In the end, $\mathrm{val}\mathcal{G}_\infty^-(v_0) = -\infty = g(v_0) = \mathrm{val}\mathcal{G}(v_0)$. As before, he cannot achieve this value in an actual play, he just can achieve arbitrarily small values by moving down to a w_i. And even these values at w_i are not actually achievable. For example, to achieve a value $\frac{1}{2}$-below $\mathcal{G}(v_0) = -\infty$, he has to get an outcome at most -2. But he cannot just move to w_1, although $\mathrm{val}\mathcal{G}(w_1) = -2$, but this is again an approximation. So to get a value $\frac{1}{2}$-below $\mathcal{G}(v_0)$, he has to move at least 5 steps to w_2 and then to one of its successor $w' \in w_2E$ with $\lambda(w') < -2$.

To prove that g is the value function of the original game, we need to introduce strategies for Player 1 and Player 0, which are inductively constructed from the strategies in the unfolding. To give an intuition for the construction, we view a play in $\mathcal{G}$ as a play in the unfolding of $\mathcal{G}$. Let us look more closely at the situation of each player.

The Strategy of Player 0

Player 0 wants to achieve the value $g_\gamma(v_0)$ or come ε-close. To reach this goal, she imagines to play in $\mathcal{G}_\gamma^-$ and uses her ε-optimal strategies $\sigma_\gamma^\varepsilon$ for that game. Between every two occurrences of nodes of minimal priority throughout the play, she plays a strategy $\sigma_\gamma^{\varepsilon_i}$.

Initially, ε_i will be $\frac{\varepsilon}{2}$, ε being the approximation value she wants to attain in

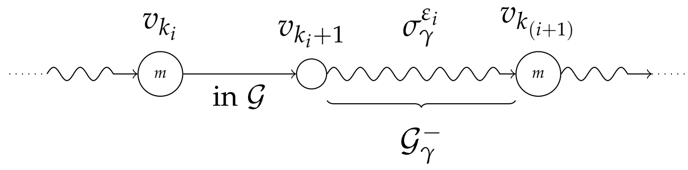


Figure 3.7: Player 0's strategy after having seen i nodes of priority m

the end. Then she chooses a smaller ε_{i+1} every time she passes an edge outside of $\mathcal{G}^-$. She will adjust the approximation value by cutting it in half every time she changes the strategy. This is illustrated in Figure 3.20.

For a history h or a full play π, let $L(h)$ (resp. $L(\pi)$) be the number of nodes with minimal priority m occurring in h (or π).

Definition 3.20. The strategy σ^ε for Player 0 in the game $\mathcal{G}$, after history $h = v_0, \ldots, v_\ell$ is given as follows. In the case that $L(h) = 0$ (i.e., no position of minimal priority has been seen), let $\varepsilon(h) := \varepsilon/2$, and $\sigma^\varepsilon(h) := \sigma_\gamma^{\varepsilon(h)}(h)$. Otherwise, let v_k be the last node of priority m in the history $h = v_0, \ldots, v_\ell$,

$$\varepsilon(h) := \frac{\varepsilon}{2^{L(h)+1}}$$

and

$$\sigma^\varepsilon(h) := \sigma_\gamma^{\varepsilon'}(v_{k+1}, \ldots, v_\ell).$$

Now let us consider a play $\pi = v_0, \ldots, v_k, v_{k+1}, \ldots$, consistent with a strategy σ^ε, where v_k is the first node with minimal priority. The following property about values $g_\gamma(v_0)$ and $g_\gamma(v_{k+1})$ in such a case (and an analogous, but more tedious one for Player 1) allows us to prove ε-optimality of the strategies σ^ε, as stated in the proposition below.

Lemma 3.21. *$g_\gamma(v_{k+1})$ is $\frac{\varepsilon}{2}$-above $g_\gamma(v_0)$.*

Proof. Let us look at π as played in $\mathcal{G}_{\gamma+1}^- = \mathcal{G}_\gamma^-$. By definition, $\lambda_{\gamma+1}(v_k) = g_\gamma(v_{k+1})$, hence

$$\mathrm{p}(\pi) = \lambda_{\gamma+1}(v_k) = g_\gamma(v_{k+1}).$$

As $\sigma_\gamma^{\frac{\varepsilon}{2}}$ is $\frac{\varepsilon}{2}$-optimal in $\mathcal{G}_\gamma^-$, we know that $\mathrm{p}(\pi)$ is $\frac{\varepsilon}{2}$-above

$$\mathrm{val}\mathcal{G}_{\gamma+1}^-(v_0) = g_{\gamma+1}(v_0) = g_\gamma(v_0).$$

□

Proposition 3.22. *The strategy σ^ε is ε-optimal, i.e. for every $v \in V$ and every strategy ρ for Player* 1, $\mathrm{p}(\pi_{\sigma^\varepsilon,\rho}(v))$ *is ε-above $g(v)$.*

Proof. Let us fix v and a strategy ρ for Player 1. We distinguish the following two cases.

Case 1: $\pi_{\sigma^\varepsilon,\rho}(v)$ visits nodes of minimal priority infinitely often.
In this case, the outcome of the play is ∞ and there is nothing left to show.

Case 2: $\pi_{\sigma^\varepsilon,\rho}(v)$ visits nodes of minimal priority only finitely often.
We prove this case by induction over the number of nodes with minimal priority occurring during the play.
If $L(\pi_{\sigma^\varepsilon,\rho}(v)) = 0$, then the whole play is equivalent to a play in $\mathcal{G}_\gamma^-$ and σ^ε is equivalent to the $\frac{\varepsilon}{2}$-optimal strategy $\sigma_\gamma^{\frac{\varepsilon}{2}}$ and thus gives a payoff $\frac{\varepsilon}{2}$-above $g_\gamma(v) = g(v)$.

Now let us look at a play $\pi_{\sigma^\varepsilon,\rho}(v) = v_0, \ldots, v_k, v_{k+1}, \ldots$, where v_k is the first node with minimal priority and $L(\pi_{\sigma^\varepsilon,\rho}(v)) = n$. For a play suffix s, let

$$\rho'(s) = \rho(v_0, \ldots, v_k, s) \text{ and } \sigma'(s) = \sigma^{\frac{\varepsilon}{2}}(s) = \sigma^\varepsilon(v_0, \ldots, v_k, s),$$

i.e. we play a part of $\pi_{\sigma^\varepsilon,\rho}$, starting at v_{k+1}.

As $L(\pi_{\sigma',\rho'}(v_{k+1})) = n - 1$, by induction hypothesis, $\mathrm{p}(\pi_{\sigma^\varepsilon,\rho'}(v_{k+1}))$ is $\frac{\varepsilon}{2}$-above $g(v_{k+1})$. By Observation 3.11 it follows, that $\mathrm{p}(\pi_{\sigma^\varepsilon,\rho'}(v_{k+1}))$ is $\frac{\varepsilon}{2}$-above $g(v_{k+1})$. Using Lemma 3.21, we get that $g_\gamma(v_{k+1})$ is $\frac{\varepsilon}{2}$-above $g_\gamma(v_0)$.

By transitivity (Observation 3.11) and the above, we obtain that

$$\mathrm{p}(\pi_{\sigma_\gamma^{\frac{\varepsilon}{2}},\rho}(v_0)) = \mathrm{p}(\pi_{\sigma^\varepsilon,\rho'}(v_{k+1})) \text{ is } \varepsilon\text{-above } g_\gamma(v_0)$$

and this concludes the proof. □

The Strategy of Player 1

Now we look at the situation of Player 1. The problem of Player 1 is that he cannot just combine his strategies for $\mathcal{G}_\gamma^-$. If he did, he would risk going infinitely often through nodes with minimal priority which is his worst case scenario. Intuitively speaking, he needs a way to count down, so that he is able to come close enough to his desired value, but stops going through the nodes with minimal priority after a finite number of times. To achieve that, he utilises the strategy index as a counter. Like Player 0, he starts with a strategy for $\mathcal{G}_\gamma^-$, but with every time he passes a node of minimal priority and

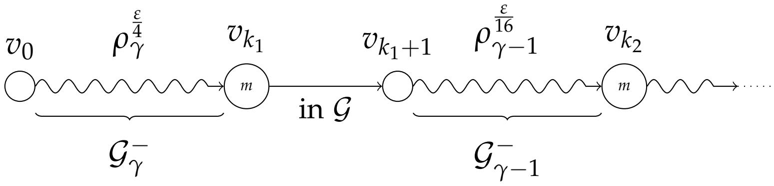


Figure 3.8: Player 1's strategy at the beginning of the play for a successor ordinal γ

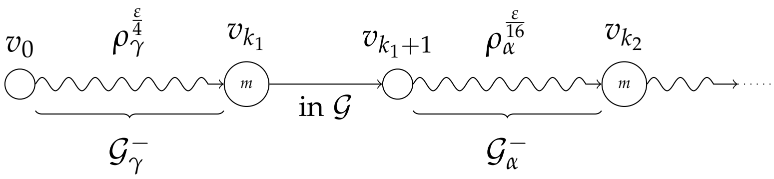


Figure 3.9: Player 1's strategy at the beginning of the play for a limit ordinal γ

changes his strategy, he not only adjusts the approximation value according to the previous one, but also lowers the strategy index in the following way. If the current game index is a successor ordinal, he just changes the index to its predecessor and adjusts the approximation value in the same way Player 0 does. In Figure 3.8, we illustrate this by an example. The nodes v_{k_i} depict the positions with minimal priority.

If the current game index is a limit value, he uses the fact that there is a game index belonging to a game which has an outcome close enough to his desired value. In the situation depicted in Figure 3.9 he would choose an α such that $\mathrm{val}\mathcal{G}_\alpha^-(v_{k_1+1})$ is $\frac{\varepsilon}{4}$-below $\lambda_\gamma(v_{k_1})$.

Finally, after a finite number of changes, as the ordinals are well-founded, he will be playing some version of $\rho_\alpha^{\varepsilon_l}$ (for a fixed ordinal α) and keep on playing this strategy for the rest of the play.

Now we formally describe Player 1's strategy. Let us first fix some notation considering game indices. For a limit ordinal α, a node $v \in V$ of priority m, and for $\varepsilon \in (0,1)$, we denote by $\alpha \restriction \varepsilon, v$ the index for which the value $\mathrm{val}\mathcal{G}_\alpha^-(v)$ is ε-below $\lambda_\alpha(w)$, where $\{w\} = vE$.

Definition 3.23. For a given approximation value ε', a starting ordinal ζ, and a history $h = v_0, \ldots, v_\ell$, we define game indices $\alpha_\zeta(h, \varepsilon')$, approximation values

$\varepsilon(h, \varepsilon')$, and a strategy $\rho^{\varepsilon'}$ for Player 1 in the following way.

If $L(h) = 0$, we fix $\alpha_\zeta(h, \varepsilon') = \zeta$ and $\varepsilon(h, \varepsilon') = \varepsilon'$.

For $h = v_0, \ldots, v_k, v_{k+1}, \ldots, v_l$, where v_k is the last node with minimal priority in h, let $h' = v_0, \ldots, v_{k-1}$ and put

$$\alpha_\zeta(h, \varepsilon') = \begin{cases} \alpha_\zeta(h', \varepsilon') - 1 & \text{for } \alpha_\zeta(h', \varepsilon') \text{ successor ordinal,} \\ \alpha_\zeta(h', \varepsilon') \restriction (\frac{\varepsilon'}{4^{L(h')+1}}, v_k) & \text{for } \alpha_\zeta(h', \varepsilon') \text{ limit ordinal,} \\ 0 & \text{for } \alpha_\zeta(h', \varepsilon') = 0, \end{cases}$$

and $\varepsilon(h, \varepsilon') = \frac{\varepsilon'}{4^{L(h)}}$.

The ε'-optimal strategy for Player 1 is given by:

$$\rho_\zeta^{\varepsilon'}(v_0 \ldots v_l) = \rho_{\alpha_\zeta(v_0 \ldots v_l, \varepsilon')}^{\frac{\varepsilon(v_0 \ldots v_l, \varepsilon')}{4}}.$$

We motivate the above definition with a nice property of a play consistent with such a strategy for Player 1. Let

$$\pi = v_0, \ldots, v_{k_1}, v_{k_1+1}, \ldots, v_{k_l}, v_{k_l+1}, \ldots$$

be a play that is consistent with a strategy $\rho_\zeta^{\varepsilon'}$, where v_{k_i} are the nodes with minimal priority.

To simplify the notation, let $k_0 = -1$, and $\alpha_0 = \zeta, \alpha_i = \alpha_\zeta(v_0, \ldots, v_{k_i}, \varepsilon')$ and $\varepsilon_i = \frac{\varepsilon'}{4^i} = \varepsilon(v_0, \ldots, v_{k_i}, \varepsilon')$ (describing the situation after i occurrences of nodes with minimal priority).

Lemma 3.24. *$g_{\alpha_{i+1}}(v_{k_{(i+1)}+1})$ is $\frac{\varepsilon_i}{2}$-below $g_{\alpha_i}(v_{k_i+1})$.*

Proof. First, we observe that by definition,

$$g_{\alpha_{i+1}}(v_{k_{(i+1)}+1}) \text{ is } \frac{\varepsilon'}{4^{i+1}} = \frac{\varepsilon_i}{4}\text{-below } \lambda_{\alpha_i}(v_{k_{(i+1)}}).$$

On the part $\pi_i = v_{k_i+1} \ldots v_{k_{(i+1)}}$ of the play, the strategy ρ_ζ^ε is equivalent to the $\frac{\varepsilon_i}{4}$-optimal strategy $\rho_{\alpha_i}^{\varepsilon_i}$ in $\mathcal{G}_{\alpha_i}^-$ and thus yields a payoff $\mathrm{p}(\pi_i)$ that is $\frac{\varepsilon_i}{4}$-below $g_{\alpha_i}(v_{k_i+1})$. By definition, $\mathrm{p}(\pi_i) = \lambda_{\alpha_i}(v_{k_{(i+1)}})$. Thus, by Observation 3.11, it follows that $g_{\alpha_{i+1}}(v_{k_{(i+1)}+1})$ is $\frac{\varepsilon_i}{2}$-below $g_{\alpha_i}(v_{k_i+1})$. □

Using the lemma above, we can now prove the ε-optimality of Player 1's strategy.

Proposition 3.25. *The strategy $\rho^{\varepsilon}_{\zeta}$ is ε-optimal, i.e. for every $\varepsilon \in (0,1)$, for all $v \in V$, and strategies σ of Player 0: $\mathrm{p}(\pi_{\sigma,\rho^{\varepsilon}_{\zeta}}(v))$ is ε-below $g_{\zeta}(v)$.*

Proof. Let us fix a strategy σ for Player 0 and distinguish two cases.

Case 1: $\pi_{\sigma,\rho^{\varepsilon}_{\zeta}}(v)$ visits nodes of minimal priority infinitely often.

In this case, we show that $g_{\zeta}(v) = \infty$. Towards a contradiction, assume that $g_{\zeta}(v) < \infty$ and consider,

$$\pi_{\sigma,\rho^{\varepsilon}_{\zeta}}(v) = v_0, \ldots, v_{k_1}, v_{k_1+1}, \ldots, v_{k_\ell}, v_{k_{\ell+1}}, \ldots,$$

where v_{k_i} are the nodes with minimal priority. Now we can use Lemma 3.24, i.e. we know for $k_0 = -1$, and $\alpha_0 = \zeta, \alpha_i = \alpha_{\zeta}(v_0, \ldots, v_{k_i}, \varepsilon)$, that

$$g_{\alpha_{i+1}}(v_{k_{(i+1)}+1}) \text{ is } \frac{\varepsilon_i}{2}\text{-below } g_{\alpha_i}(v_{k_i+1}).$$

In particular, if $g_{\alpha_i}(v_{k_i+1})$ is finite, then $g_{\alpha_{i+1}}(v_{k_{(i+1)}+1})$ is finite as well, as ∞ is only ε-close to ∞ (see the remark after the definition of ε-closeness).

The sequence $\alpha_0 > \alpha_1 > \ldots$ is a strictly decreasing sequence of ordinals and therefore, for some l, $\alpha_l = 0$. But in the game $\mathcal{G}_0^-$, node v_{k_l} is a terminal node, and thus $g_{\alpha_l}(v_{k_l}) = g_0(v_{k_l}) = \lambda_0(v_{k_l}) = \infty$, which, using the above, contradicts $g_{\zeta}(v) < \infty$.

Case 2: $\pi_{\sigma,\rho^{\varepsilon}_{\zeta}}(v)$ visits nodes of minimal priority only finitely often.

We will prove this case by induction over the number of nodes with minimal priority.

If $L(\pi_{\sigma,\rho^{\varepsilon}_{\zeta}}(v)) = 0$ then the whole play is equivalent to a play in $\mathcal{G}_{\zeta}$ and $\rho^{\varepsilon}_{\zeta}$ is equivalent to the $\frac{\varepsilon}{4}$-optimal strategy $\rho^{\frac{\varepsilon}{4}}_{\zeta}$ and thus gives a payoff $\frac{\varepsilon}{4}$-below $g_{\zeta}(v)$. Now let us look at a play

$$\pi_{\sigma,\rho^{\varepsilon}_{\zeta}}(v) = v_0, \ldots, v_k, v_{k+1}, \ldots,$$

where v_k is the first node with minimal priority and $L(\pi_{\sigma,\rho^{\varepsilon}_{\zeta}}(v)) = n$.

For a play suffix s, let $\sigma'(s) = \sigma(v_0, \ldots, v_k, s)$ and

$$\rho'(s) = \rho^{\frac{\varepsilon}{4}}_{\alpha_1}(s) = \rho^{\varepsilon}_{\alpha_0=\zeta}(v_0 \ldots v_k s).$$

As $L(\pi_{\sigma',\rho'}(v_{k+1})) = n - 1$, by induction hypothesis, $\mathrm{p}(\pi_{\sigma',\rho'}(v_{k+1}))$ is $\frac{\varepsilon}{4}$-below $g_{\alpha_1}(v_{k+1})$.

By Observation 3.11, $\mathrm{p}(\pi_{\sigma',\rho'}(v_{k+1}))$ is $\frac{\varepsilon}{4}$-below $g_{\alpha_1}(v_{k+1})$.

Now we can use Lemma 3.24, which tells us, that $g_{\alpha_1}(v_{k+1})$ is $\frac{\varepsilon}{2}$-below $g_{\alpha_0}(v_0)$.

By Observation 3.11, we get that

$$\mathrm{p}(\pi_{\sigma,\rho^{\varepsilon}_{\zeta}(v_0)}) = \mathrm{p}(\pi_{\sigma',\rho'}(v_{k+1})) \text{ is } \frac{\varepsilon}{4} + \frac{\varepsilon}{2} < \varepsilon\text{-below } g_{\zeta}(v_0),$$

which completes the proof. □

Having defined the ε-optimal strategies σ^{ε} and $\rho^{\varepsilon}_{\gamma}$, we can conclude the following.

Proposition 3.26. *For a quantitative parity game $\mathcal{G} = (V, E, \lambda, \Omega)$, for all $v \in V$,*

$$\sup_{\sigma\in\Gamma_0}\inf_{\rho\in\Gamma_1} \mathrm{p}(\pi_{\sigma,\rho}(v)) = \inf_{\rho\in\Gamma_1}\sup_{\sigma\in\Gamma_0} \mathrm{p}(\pi_{\sigma,\rho}(v)) = \mathrm{val}\mathcal{G}(v) = g(v).$$

Correctness of Model-Checking Games

After establishing determinacy for quantitative parity games we are ready to prove the second part of Theorem 3.7, namely that the model-checking games correctly describe the evaluation of a Qμ formula. Let us first look at formulae without fixed-point operators and prove the following lemma.

Lemma 3.27. MC$[\mathcal{Q}, \varphi]$ *is a model-checking game for $\varphi \in$ QML.*

Proof. Note that in case of QML-model checking all plays are finite and the game graph is a tree of finite depth. The value of a the game from node v is $f(v)$ as defined in Section 3.3. Note that the process of computing the sequence f_i stops after a finite number of steps, namely the depth of the game tree. We have to show by induction on the structure of φ that indeed $f(v) = [\![\varphi]\!](v)$.

In case that $\varphi = P_i + c$ or $\varphi = \neg(P_i + c)$ (formulae are assumed to be in negation normal form), the value of the formula is the same as the evaluation of the terminal position by definition. In case that $\varphi = \varphi_1 \wedge \varphi_2$, we have the evaluation $[\![\varphi]\!](v) = \min\{[\![\varphi_1]\!](v), [\![\varphi_2]\!](v)\}$. But that also means, that in the model-checking game, the corresponding position belongs to Player 1 and the value $f(v)$ is computed as $\min_{w\in vE} f(w)$, where the next positions w are the subformulae φ_1, φ_2, evaluated at v. By induction hypothesis, the values $f(w)$ coincide with $[\![\varphi_1]\!](v)$ and $[\![\varphi_2]\!](v)$.

In case that $\varphi = \Box\varphi'$, we have $[\![\varphi]\!](v) = \inf_{w\in vE}[\![\varphi']\!](w)$. Hence, in the model-checking game, the corresponding position belongs to Player 1 and the value

$f(v)$ is computed as $\inf_{w \in vE} f(w)$, where the next positions are the subformula φ' evaluated at each of the successor nodes w in the original transition system. By induction hypothesis, these values coincide with $[\![\varphi']\!](w)$. The proofs for $\varphi = \varphi_1 \vee \varphi_2$ and $\varphi = \Diamond\varphi'$ are analogous, only that the corresponding positions now belong to Player 0. $\square$

To prove the correctness for Qμ, we need to consider formulae with fixed-point operators. Let us consider the case that $\varphi = \nu X.\psi$.

Note that in the game $\mathrm{MC}[\mathcal{Q}, \varphi]$, the positions with minimal priority are of the form (X, v) each with a unique successor (φ, v). Our induction hypothesis states that for every interpretation g of the fixed-point variable X, it holds that:

$$[\![\varphi]\!]^{\mathcal{Q}}_{[X\leftarrow g]} = \mathrm{valMC}[\mathcal{Q}, \psi[X/g]]. \tag{3.1}$$

By Theorem 2.6, we know that we can compute $\nu X.\psi$ inductively in the following way: $[\![\nu X.\psi]\!]^{\mathcal{K}}_{\mathfrak{I}} = g_\gamma$ with $g_0(v) = \infty$ for all $v \in V$ and

$$g_\alpha = \begin{cases} [\![\psi]\!]_{\varepsilon[X\leftarrow g_{\alpha-1}]} & \text{for } \alpha \text{ successor ordinal,} \\ \lim_{\beta<\alpha} [\![\psi]\!]_{\varepsilon[X\leftarrow g_\beta]} & \text{for } \alpha \text{ limit ordinal,} \end{cases}$$

and where $g_\gamma = g_{\gamma+1}$.

Now we want to prove that the games $\mathrm{MC}[\mathcal{Q}, \psi[X/g_\alpha]]$ coincide with the unfolding of $\mathrm{MC}[\mathcal{Q}, \varphi]$. We say that two games coincide if the game graph is essentially the same, except for some additional moves where neither player has an actual choice. In our case these are the moves from $\varphi = \nu X.\psi$ to ψ, which allows us to show the following lemma.

Lemma 3.28. *The games* $\mathrm{MC}[\mathcal{Q}, \psi[X/g_\alpha]]$ *and* $\mathrm{MC}[\mathcal{Q}, \varphi]^-_\alpha$ *coincide for all* α.

Proof. Considering $\alpha = 0$, note that $\mathrm{MC}[\mathcal{Q}, \psi[X/g_0]]$ coincides with $\mathrm{MC}[\mathcal{Q}, \varphi]^-_0$ by construction. The induction hypothesis is that $\mathrm{MC}[\mathcal{Q}, \psi[X/g_\alpha]]$ coincides with $\mathrm{MC}[\mathcal{Q}, \varphi]^-_\alpha$ for some α.

If α is a successor ordinal: the interesting positions in $\mathrm{MC}[\mathcal{Q}, \psi[X/g_\alpha]]$ are the terminal positions of the form (g_α, a) with valuation

$$\lambda(g_\alpha, a) = g_\alpha(a) = [\![\psi]\!]_{[X\leftarrow g_{\alpha-1}]}(a).$$

In $\mathrm{MC}[\mathcal{Q}, \varphi]^-_\alpha$, the corresponding terminal positions are of the form (X, a) with valuation

$$\lambda_\alpha(X, a) = \mathrm{valMC}[\mathcal{Q}, \varphi]^-_{\alpha-1}(\varphi, a).$$

By induction hypothesis, we have $\mathrm{MC}[\mathcal{Q}, \varphi]_{\alpha-1}$ coincides with $\mathrm{MC}[\mathcal{Q}, \psi[X/g_{\alpha-1}]]$ and therefore, by Equation 3.1,

$$\mathrm{valMC}[\mathcal{Q}, \varphi]^{-}_{\alpha-1}(\varphi, a) = \mathrm{valMC}[\mathcal{Q}, \psi[X/g_{\alpha-1}]](\psi, a) = [\![\psi]\!]_{[X \leftarrow g_{\alpha-1}]}(a).$$

If α is a limit ordinal: we only consider terminal positions in $\mathrm{MC}[\mathcal{Q}, \psi[X/g_{\alpha}]]$ of the form (g_α, a) with valuation

$$\lambda(g_\alpha, a) = g_\alpha(a) = \lim_{\beta<\alpha} [\![\psi]\!]_{[X \leftarrow g_\beta]}(a).$$

In $\mathrm{MC}[\mathcal{Q}, \varphi]^{-}_{\alpha}$, the corresponding terminal positions are of the form (X, a) with valuation

$$\lambda_\alpha(X, a) = \lim_{\beta<\alpha} \mathrm{valMC}[\mathcal{Q}, \varphi]^{-}_{\beta}(\varphi, a).$$

By induction hypothesis, we have that for all $\beta < \alpha$ that $\mathrm{MC}[\mathcal{Q}, \varphi]_\beta$ coincides with $\mathrm{MC}[\mathcal{Q}, \psi[X/g_\beta]]$ and therefore, by Equation 3.1,

$$\lim_{\beta<\alpha} \mathrm{valMC}[\mathcal{Q}, \varphi]^{-}_{\beta}(\varphi, a) = \lim_{\beta<\alpha} \mathrm{valMC}[\mathcal{Q}, \psi[X/g_\beta]](\psi, a) = \lim_{\beta<\alpha} [\![\psi]\!]_{[X \leftarrow g_\beta]}(a).$$

For $\varphi = \mu X.\psi$ the proof is analogous. □

From the above lemma and Proposition 3.26, we conclude that the value of the game $\mathrm{MC}[\mathcal{Q}, \varphi]$ is the limit of the values $\mathrm{MC}[\mathcal{Q}, \varphi]^{-}_{\alpha}$, whose value functions coincide with the stages of the fixed-point evaluation g_α for all α, and thus

$$\mathrm{valMC}[\mathcal{Q}, \varphi] = \mathrm{valMC}[\mathcal{Q}, \varphi]^{-}_{\gamma} = g_\gamma = [\![\varphi]\!]^{\mathcal{Q}}.$$

This concludes the proof of Theorem 3.7.

Positional ε-Optimal Strategies

After establishing determinacy for quantitative parity games, we now come back to positional determinacy. As shown in the examples in Section 3.1, quantitative parity games do not admit optimal strategies. However, they admit ε-optimal strategies and are even positionally determined in such strategies.

Theorem 3.29. *For a quantitative parity game $\mathcal{G} = (V, V_0, V_1, E, \lambda, \Omega)$ and a fixed $\varepsilon \in (0, 1)$, Player 0 has a positional ε-optimal strategy σ, i.e. for every strategy ρ of Player 1, $\mathrm{p}(\pi_{\sigma,\rho}(v))$ is ε-close to $\mathrm{val}\mathcal{G}(v)$.*

Proof. For a quantitative parity game $\mathcal{G} = (V, V_0, V_1, E, \lambda, \Omega)$, for every $r \in \mathbb{R}_\infty$, let $\mathcal{G}_r$ be the classical parity game where Player 0 wins a finite play if she would get a payoff greater or equal to r in the corresponding play in the original game. She wins an infinite play if she would get a payoff of ∞ accordingly. Formally, using the convention that the player who cannot move at a position loses, we define $\mathcal{G}_r = (V, V_{0r}, V_{1r}, E, \Omega)$ where

$$V_{0r} = \{v \in V_0 \mid vE \neq \emptyset\} \cup \{v \in V \mid vE = \emptyset \wedge \lambda(v) < r\} \text{ and}$$

$$V_{1r} = \{v \in V_1 \mid vE \neq \emptyset\} \cup \{v \in V \mid vE = \emptyset \wedge \lambda(v) \geq r\}.$$

As we have mentioned before, classical parity games are positionally determined. This means that in every game $\mathcal{G}_r$, if Player 0 has a winning strategy from a node v, she also has a positional one. Now given a quantitative parity game $\mathcal{G}$ and $\varepsilon \in (0,1)$, for a position v, we look at $\mathcal{G}_r$ where r is a value ε-close to $\text{val}\mathcal{G}(v)$. Player 0 has an ε-optimal strategy in $\mathcal{G}$ which is a winning strategy in $\mathcal{G}_r$. This means, she also has a positional winning strategy in $\mathcal{G}_r$ and thus a positional strategy in $\mathcal{G}$ to get outcome r, i.e. a positional ε-optimal strategy. □

3.4 Bisimulation via Model Checking

Now we show a nice application of model-checking games. In Section 2.3, we proved that the quantitative μ-calculus is invariant under quantitative bisimulation by an induction on the structure of the formula and using the relational description of bisimulation. Here, we instead use the description of the value of a formula by a model-checking game and the bisimulation distance by the corresponding bisimulation game. The proof then works by playing two model-checking games and one bisimulation game at once.

Theorem 3.30 (2.18). *Let $\varphi \in \mathrm{Q}\mu$ and $v \in \mathcal{M}, v' \in \mathcal{M}'$ with $\text{val}\mathcal{G}_\sim(\mathcal{M}, v, \mathcal{M}', v') = r$ then $|[\![\varphi]\!]^{\mathcal{M}}(v) - [\![\varphi]\!]^{\mathcal{M}'}(v')| \leq r$ or, in other words,*

$$(\mathcal{M}, v) \sim^r (\mathcal{M}', v') \textit{ implies } (\mathcal{M}, v) \equiv^r_{\text{QML}} (\mathcal{M}', v').$$

Proof. Towards a contradiction assume that $\text{val}\mathcal{G}_\sim(\mathcal{M}, v, \mathcal{M}', v') = r$, but there is a formula $\varphi \in \mathrm{Q}\mu$, such that $[\![\varphi]\!]^{\mathcal{M}}(v) = k$ and $[\![\varphi]\!]^{\mathcal{M}'}(v') = k'$ and $|k' - k| > r$ (without loss of generality we assume $k' > k$).

By assumption (and Theorem 2.15), Duplicator has a positional strategy β in the bisimulation game $\mathcal{G}_{\sim}(\mathcal{M}, v, \mathcal{M}', v')$ such that

$$\sup_{\gamma \in \Gamma} \mathrm{p}(\pi_{\beta,\gamma}(v, v)) \leq r,$$

where Γ denotes the set of strategies of Spoiler. This means that for all $(w, w') \in \pi_{\beta,\gamma}(v, v')$ we have $\mathrm{pd}(w, w') \leq r$.

Furthermore, by Theorem 3.7, we know that if $[\![\varphi]\!]^{\mathcal{M}}(v) = p$ then Verifier has a strategy σ in $\mathrm{MC}(\mathcal{M}, \varphi)$ from node (φ, v) such that

$$\inf_{\rho \in \Gamma_1} \mathrm{p}(\pi_{\sigma,\rho}(v)) \text{ is } \varepsilon\text{-above } p$$

and that Falsifier has a strategy ρ in $\mathrm{MC}(\mathcal{M}, \varphi)$ such that

$$\sup_{\sigma \in \Gamma_0} \mathrm{p}(\pi_{\sigma,\rho}(v)) \text{ is } \varepsilon\text{-below } p,$$

where Γ_0 and Γ_1 denote the sets of strategies of Verifier and Falsifier.

First, let us assume that $k, k' \neq \pm\infty$. Then, we know from the above that Verifier has a strategy σ in $\mathrm{MC}(\mathcal{M}, \varphi)$ from node (φ, v) such that

$$\inf_{\rho \in \Gamma_1} \mathrm{p}(\pi_{\sigma,\rho}(v)) \geq k - \varepsilon$$

and that Falsifier has a strategy ρ in $\mathrm{MC}(\mathcal{M}', \varphi)$ such that

$$\sup_{\sigma \in \Gamma_0} \mathrm{p}(\pi_{\sigma,\rho}(v')) \leq k' + \varepsilon.$$

We consider the case in which both model-checking games and the bisimulation game are played in parallel according to these strategies, resulting in the plays $\pi_{\sigma,\rho'}(v)$ and $\pi_{\sigma',\rho}(v')$. The counter strategies ρ' and σ' will be constructed according to the winning strategies in the other game using the bisimulation relation.

We start at positions (v, φ) and (v', φ) with $\mathcal{M}, v \sim_r \mathcal{M}', v'$. In every step of our two games, after history $H = (v_0, v'_0) \ldots (v_n, v'_n)$ (we abbreviate with $h = v_0, v_1 \ldots$ and $h' = v'_0, v'_1, \ldots$ the histories in the model-checking games), we choose the next positions according to the following rules:

$$
\begin{tikzcd}
\mathrm{MC}[\mathcal{Q},\varphi] & & \mathrm{MC}[\mathcal{Q}',\varphi] \\
(v,\varphi) \arrow[d, rightsquigarrow] & & (v',\varphi) \arrow[d, rightsquigarrow] \\
(u,\psi) \in V_0 \arrow[d, "\sigma"] & \rightsquigarrow & (u',\psi) \in V_0 \arrow[d] \\
(w,\vartheta) & u = w \ \rightsquigarrow & (u',\vartheta)
\end{tikzcd}
$$

Figure 3.10: Player 0 position in the model-checking game, case 1

$$
\begin{tikzcd}
\mathrm{MC}[\mathcal{Q},\varphi] & & \mathrm{MC}[\mathcal{Q}',\varphi] \\
(v,\varphi) \arrow[d, rightsquigarrow] & & (v',\varphi) \arrow[d, rightsquigarrow] \\
(u,\psi) \in V_0 \arrow[d, "\sigma"] & \rightsquigarrow & (u',\psi) \in V_0 \arrow[d] \\
(w,\vartheta) & u \neq w \ \rightsquigarrow & (w',\vartheta)
\end{tikzcd}
$$

with $\beta(\ldots,(w,\cdot)) = (w,w')$

Figure 3.11: Player 0 position in the model-checking game, case 2

$$
\begin{tikzcd}
\mathrm{MC}[\mathcal{Q},\varphi] & \mathrm{MC}[\mathcal{Q}',\varphi] \\
(v,\varphi) \arrow[d, squiggly] & (v',\varphi) \arrow[d, squiggly] \\
(u,\psi) \in V_1 \arrow[d] & (u',\psi) \in V_1 \arrow[d, "\gamma"] \\
(w,\vartheta) \text{ analogous} & (w',\vartheta)
\end{tikzcd}
$$

Figure 3.12: Player 1 position in the model-checking game

If $(v,\varphi) \in V_0$, (i.e. also $(v',\varphi) \in V_0$, since by definition this is decided only by the φ-part of the position) we choose a move according to σ in $\mathrm{MC}(\mathcal{M},\varphi)$: from (v,φ) to $\sigma(hv,\varphi) = (w,\psi)$.

If $v = w$ (i.e. only the formula-part changes), we move in $\mathrm{MC}(\mathcal{M}',\varphi)$ to (v',ψ), as illustrated in Figure 3.10.

If $v \neq w$ (i.e. $\varphi = \Diamond\psi$), we move in $\mathrm{MC}(\mathcal{M}',\varphi)$ to $\beta((v,\varphi),\_)) = (w',\psi)$, as illustrated in Figure 3.11.

If $(v',\varphi) \in V_1$, analogously, we choose a move according to ρ in $\mathrm{MC}(\mathcal{M}',\varphi)$ to (w',ψ). If $v' = w'$, we move in the other game to (v,ψ). If $v \neq w$, (i.e. $\varphi = \Box\psi$) we move in $\mathrm{MC}(\mathcal{M},\varphi)$ to $\beta((\_,(v',\varphi')) = (w',\psi)$ (Figure 3.12).

Note that for the plays

$$\pi_{\sigma,\rho'} = (v_0,\varphi_0),(v_1,\varphi_1),\ldots \text{ and } \pi_{\sigma',\rho} = (v'_0,\varphi'_0),(v'_1,\varphi'_1),\ldots$$

we have that $v_i \sim_r v'_i$ and $\varphi_i = \varphi'_i$ for all i. This also implies, that either both plays are finite and have the same length or both plays are infinite. Let us see who wins in both cases.

First Case: Both plays are finite. For a play in a model-checking game to be finite it has to end in a position (v,φ) with $\varphi = P$ for a proposition P. Let us denote the end positions with (v,P) and (v',P'). Since we know that $P = P'$ and $v \sim_r v'$, this means that by definition $\mathrm{pd}(v,v') \leq r$. We can conclude that the payoffs in both plays, which are equal to $P(v)$ and $P(v')$, differ by at most r, $|P(v) - P(v')| \leq r$. However, we also know that the play in $\mathrm{MC}[\mathcal{M},\varphi]$ was played according to a strategy for Verifier that guarantees a payoff $p \geq k - \varepsilon$. The play in $\mathrm{MC}[\mathcal{M}',\varphi]$ was played according to a strategy that guarantees a payoff $p' \leq k' + \varepsilon$ for Falsifier. It follows that

$$|p' - p| \geq |k + \varepsilon - (k - \varepsilon)| > |r + 2\varepsilon| > r$$

which is a contradiction.

Second Case: Both plays are infinite. For a model-checking game to allow infinite plays is has to contain a fixed-point position and one of the following has to be the case. Either the underlying structure itself was infinite and the model-checking game is therefore infinite or the model-checking game contains a loop. In both cases there will a number of fixed-point positions visited infinitely often. We established that all (sub)formulae occurring in positions of the plays are the same and thus the same priorities occur as they are determined by the formula-part of a position only. This also means that the minimal priority occurring infinitely often is the same for both play and thus the payoffs are the same which is again a contradiction.

We are left to show that the same holds if k and k' are infinite. This means that $k = -\infty$ and $k' = \infty$, as we assumed they are different and $k' > k$. Then, we just have to choose the strategies accordingly, we now choose Verifier's strategy for $\mathrm{MC}(\mathcal{M}', \varphi)$ and Falsifier's strategy for $\mathrm{MC}(\mathcal{M}, \varphi)$. Let σ be the strategy for Verifier in $\mathrm{MC}(\mathcal{M}', \varphi)$ from position (φ, v') such that

$$\inf_{\rho \in \Gamma_1} \mathrm{p}(\pi_{\sigma,\rho}(v')) \geq \frac{1}{\varepsilon}.$$

Let ρ be the strategy for Falsifier in $\mathrm{MC}(\mathcal{M}, \varphi)$ from position (φ, v) such that

$$\sup_{\sigma \in \Gamma_0} \mathrm{p}(\pi_{\sigma,\rho}(v)) \leq -\frac{1}{\varepsilon}.$$

Then, we can use the same technique as above and, using the bisimulation strategy, construct counter strategies for the other players, respectively. We have to consider two cases again, either both resulting plays are finite or both are infinite. In the case that the plays are infinite, the situation is the same as above. In the case that both plays are finite, we end up again in positions (P, v) and (P, v') that are r-bisimilar and thus it follows that the resulting payoffs differ by at most r, $|P(v') - P(v)| \leq r$. Again, we have played one play according to Verifier's strategy in $\mathrm{MC}(\mathcal{M}', \varphi)$ which guarantees her a payoff $p' > \frac{1}{\varepsilon}$ and one according to Falsifier's strategy in $\mathrm{MC}(\mathcal{M}, \varphi)$ which guarantees him a payoff $p < -\frac{1}{\varepsilon}$. If we choose $\varepsilon > \frac{2}{r}$, we get a contradiction again, as then

$$|p' - p| > |\frac{1}{\varepsilon} - (-\frac{1}{\varepsilon})| > |\frac{2}{\varepsilon}| > r.$$

□

4 Qμ on Discounted Systems

In this chapter, we discuss an extension of the quantitative μ-calculus tailored for discounted transition systems, i.e. quantitative transition systems where additionally the edges are labelled.

This version of Qμ was the first one we defined in [16, 17]. It is based on a logic introduced in [9, 10] by de Alfaro, Faella, and Stoelinga. They introduce a quantitative extension of the modal μ-calculus which is interpreted over metric transition systems. These are transition systems where predicates can take values from arbitrary metric spaces. They also allow discount factors in modalities that take values from the interval $(0, 1]$. The logic they study lacks a negation operator and thus the corresponding duality properties. The authors study this logic in connection with quantitative versions of bisimulation, a topic that we discussed in Section 2.2.

We take their basic idea of a quantitative extension of the modal μ-calculus but modify it in the following ways. First of all, we restrict ourselves to real numbers instead of arbitrary metric spaces. Second, we decouple the discounts from modalities and allow them to occur as factors in the formulae. We also do not require them to be less than 1 anymore. Furthermore, we modify the systems and allow edges to be labelled with discounts as well. These changes make the logic more robust and more general, and, as we show in Section 4.2, permit us to introduce a negation operator with the desired duality properties that are fundamental to a game-based analysis.

To tackle the model-checking problem in this setting, we introduce a discounted version of quantitative parity games where now also the moves are labelled with discounts. This makes the calculation of payoffs for finite plays more complicated than in the non-discounted case and has some consequences. Discounted parity games do not retain the nice properties that classical parity games admit. Namely, they do not enjoy optimal strategies, unsurprisingly as their non-discounted counterparts also do not have this property. However, quantitative parity games at least admit positional ε-optimal strategies, discounted parity games unfortunately lose this property. In the discounted case, there are simple games where the players do not have positional ε-optimal strategies anymore. Even worse, there are games where they do not even have bounded-memory ε-optimal strategies for a fixed ε. Nevertheless, the correctness of the model-checking theorem follows from the correctness in the non-discounted case, as we show later.

We briefly discuss another quantitative version of parity games, crash games that were introduced by Gawlitza and Seidl in [20]. These games are essentially the same as finite quantitative parity games over integers. However, Gawlitza and Seidl do not study crash games in connection to a logic, but instead show how to solve these games by a reduction to hierarchical systems and using a variant of strategy improvement. We show that the algorithm they present can also be used to solve finite discounted (and non-discounted) quantitative parity games over integers.

In the final section of this chapter, we show that in the discounted case, we can also establish the other direction of the classical model-checking theorem. Namely, we define the value of a discounted parity game (with a fixed number of priorities) by a formula of the discounted quantitative μ-calculus.

Please note that we present a multiplicative version of the logic here, in contrast to the version with addition in Chapter 2. However, as we demonstrate later, we can easily transform the semantics of formulae (and the payoff of games) from the additive version to the multiplicative version and back using a logarithm function.

4.1 Syntax and Semantics of Discounted Qμ

In this section, we introduce discounted quantitative transition systems and the discounted quantitative μ-calculus dQμ. We present a multiplicative version of the calculus and we use only the non-negative reals as domain for our quantities. As before, $\mathbb{R}^+$ is the set of positive real numbers, $\mathbb{R}^+_0 := \mathbb{R}^+ \cup \{0\}$, and $\mathbb{R}^+_\infty := \mathbb{R}^+_0 \cup \{\infty\}$. Discounted quantitative transition systems are an extension of quantitative transition systems; they not only allow quantitative predicates at nodes, but also discounts (or weights) on edges.

Definition 4.1. A *discounted transition system (DTS)* is a tuple

$$\mathcal{Q} = (V, E, \delta, \{P_i\}_{i \in I}),$$

consisting of a directed graph (V, E), a discount function $\delta : E \to \mathbb{R}^+$ that assigns a value to each edge, and predicate functions $P_i : V \to \mathbb{R}^+_\infty$ that assign values to nodes.

A discounted quantitative transition system that has $\delta(e) = 1$ for all $e \in E$ is called *non-discounted* (and thus equivalent to a quantitative transition system).

We now introduce a discounted version of Qμ to describe properties of discounted transition systems.

Definition 4.2. Given a set $\mathcal{X}$ of variables X, predicate functions $\{P_i\}_{i \in I}$, discount factors $d \in \mathbb{R}^+$ and constants $c \in \mathbb{R}^+$, the formulae of *discounted* Qμ (dQμ) are built in the following way:

(1) $|P_i - c|$ is a dQμ-formula,

(2) X is a dQμ-formula,

(3) if φ, ψ are dQμ-formulae, then so are $(\varphi \wedge \psi)$ and $(\varphi \vee \psi)$,

(4) if φ is a dQμ-formula, then so are $\Box\varphi$ and $\Diamond\varphi$,

(5) if φ is a dQμ-formula, then so is $d \cdot \varphi$,

(6) if φ is a formula of dQμ, then $\mu X.\varphi$ and $\nu X.\varphi$ are formulae of dQμ.

We use $|P_i - c|$ because we need the distance of a predicate to a constant later in Section 4.5, and we do not introduce negation here (cf. Section 4.2).

Again, for a discounted transition system $\mathcal{Q}$, we have a lattice $(\mathcal{F}, \leq)$ of functions $\mathcal{F} := \{f : V \to \mathbb{R}^+_\infty\}$ with $f = \infty$ as top element and $f = 0$ as bottom element, cf. Section 2.1.

Definition 4.3. Given a DTS $\mathcal{Q} = (V, E, \delta, \{P_i\}_{i \in I})$ and an interpretation $\mathfrak{I} : \mathcal{X} \to \mathcal{F}$, a dQ$\mu$-formula yields a valuation function $[\![\varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}} : V \to \mathbb{R}^+_\infty$ defined as follows, for every $v \in V$.

(1) $[\![|P_i - c|]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v) = |P_i(v) - c|$,

(2) $[\![X]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v) = \mathfrak{I}(X)(v)$,

(3) $[\![\varphi_1 \wedge \varphi_2]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v) = \min\{[\![\varphi_1]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v), [\![\varphi_2]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v)\}$,
$[\![\varphi_1 \vee \varphi_2]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v) = \max\{[\![\varphi_1]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v), [\![\varphi_2]\!]^{\mathcal{Q}}(v)\}$,

(4) $[\![\Diamond\varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v) = \sup_{v' \in vE} \delta(v, v') \cdot [\![\varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v')$,
$[\![\Box\varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v) = \inf_{v' \in vE} \frac{1}{\delta(v,v')} [\![\varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v')$,

(5) $[\![d \cdot \varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v) = d \cdot [\![\varphi]\!]^{\mathcal{Q}}_{\mathfrak{I}}(v)$,

(6) $[\![\mu X.\varphi]\!]_{\mathfrak{I}}^{\mathcal{Q}}(v) = \inf\{f \in \mathcal{F} : f = [\![\varphi]\!]_{\mathfrak{I}[X\leftarrow f]}^{\mathcal{Q}}\}(v)$,
$[\![\nu X.\varphi]\!]_{\mathfrak{I}}^{\mathcal{Q}}(v) = \sup\{f \in \mathcal{F} : f = [\![\varphi]\!]_{\mathfrak{I}[X\leftarrow f]}^{\mathcal{Q}}\}(v)$.

We extend the muliplication from $\mathbb{R}_0^+$ to $\mathbb{R}_\infty^+$ in the following way, let $a, b \in \mathbb{R}_\infty^+$ and we assume without loss of generality that $a \geq b$, then we define

$$a * b = \begin{cases} a * b & \text{if} \quad a \in \mathbb{R}_0^+ \text{ and } b \in \mathbb{R}_0^+ \\ \infty & \text{if} \quad a = \infty \text{ and } b \geq 0 \\ 0 & \text{if} \quad a = \infty \text{ and } b = 0 \end{cases}$$

As before, all operators are monotone and thus we use the Knaster-Tarski theorem to establish that the fixed points exist and can be computed inductively as described in Theorem 2.6 (for the least fixed point, we now start with the function that assigns 0 to every node instead of $-\infty$).

The fragment of dQμ consisting of formulae without fixed-point operators is called *discounted quantitative modal logic* dQML. Again, if dQμ is interpreted over qualitative transition systems, it coincides with the classical μ-calculus. Please note that here qualitative means that we only assign the values 0 and ∞. We do not have to worry about the discounts in this case as the value of a formula cannot be changed by discounting.

Over non-discounted quantitative transition systems, the definition above coincides with the one in [10]. So far, we have not discussed negation in this setting. The next task is thus to find a meaningful way to introduce a negation operator for our logic. Please note that for the semantics on discounted systems we take the natural definition for the $\Diamond$-operator and use a dual one for the $\Box$-operator, thus we have a factor $\frac{1}{\delta}$ instead of δ. We show that if we want duality of operators and use multiplication as discount operation, this is the only definition for which there is a well-behaved negation operator.

First, we illustrate by an example how the evaluation differs from the non-discounted Qμ we defined in Chapter 2.

Example 4.4. In Figure 4.1, we see two simple discounted transition systems – actually only $\mathcal{Q}'$ really makes use of the discounting feature, $\mathcal{Q}$ is non-discounted. To illustrate what effect discounting has on the evaluation of formulae, consider the formulae $\varphi = \mu X.(P \vee 2 \cdot \Diamond X)$ on $\mathcal{Q}$ and $\psi = \mu X.(P \vee \Diamond X)$ on $\mathcal{Q}'$. Again only φ makes use of discounting, whereas ψ does not. If we evaluate φ on $\mathcal{Q}$ and ψ on $\mathcal{Q}'$, we get the same results. In both evaluations we have to take the discount factor 2 into account, although it comes from different

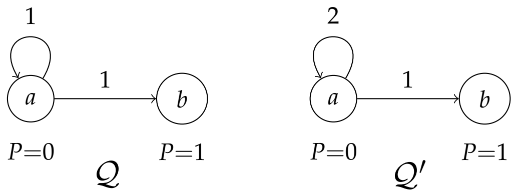


Figure 4.1: Two simple discounted transition systems

sources, once from the systems and once from the formula. The value of both formulae is ∞, $[\![\mu X.(P \vee 2 \cdot \Diamond X)]\!]^{\mathcal{Q}}(a) = [\![\mu X.(P \vee \Diamond X)]\!]^{\mathcal{Q}'}(a) = \infty$, because the discounts accumulate in each step of the fixed point evaluation.

4.2 Negation Operators

So far, the quantitative logics dQμ and dQML lack a negation operator and the associated dualities between $\wedge$ and $\vee$, $\Diamond$ and $\Box$, and between least and greatest fixed points.

Let us clarify what we expect from such an operator. Syntactically, we want to add to the formula building rules of dQμ a new rule saying that, for every formula $\varphi \in$ dQμ, also $\neg\varphi$ is a formula of dQμ. For fixed point formulae $\mu X.\varphi$ and $\nu X.\varphi$ we then have to require that X only occur positively (i.e. under an even number of negations) in φ, to guarantee monotonicity and, accordingly, the existence of least and greatest fixed points. Semantically, the meaning of negation has to be defined by an operator $f_\neg : \mathbb{R}^+_\infty \to \mathbb{R}^+_\infty$ satisfying the properties outlined in the following definition.

Definition 4.5. A *negation operator* $f_\neg$ for dQμ is a function $\mathbb{R}^+_\infty \to \mathbb{R}^+_\infty$, such that, when we define $[\![\neg\varphi]\!] = f_\neg([\![\varphi]\!])$, the following equivalences hold for every $\varphi \in$ dQμ:

(1) $\neg\neg\varphi \equiv \varphi$

(2) $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$ and $\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$

(3) $\neg\Box\varphi \equiv \Diamond\neg\varphi$ and $\neg\Diamond\varphi \equiv \Box\neg\varphi$

(4) $\neg d \cdot \varphi \equiv \beta(d) \cdot \neg\varphi$ for some $\beta : \mathbb{R}^+ \to \mathbb{R}^+$ independent of φ

(5) $\neg\mu X.\varphi \equiv \nu X.\neg\varphi[X/\neg X]$ and $\neg\nu X.\varphi \equiv \mu X.\neg\varphi[X/\neg X]$

A straightforward calculation as carried out below shows that the function

$$f_{\frac{a}{x}} : \mathbb{R}^+_\infty \to \mathbb{R}^+_\infty : x \mapsto \begin{cases} a/x & \text{for } x \neq 0, x \neq \infty, \\ \infty & \text{for } x = 0, \\ 0 & \text{for } x = \infty, \end{cases}$$

is a negation operator for dQμ.

Proposition 4.6. *$f_{\frac{a}{x}}$ is a negation operator in* dQμ *for every $a \in \mathbb{R}^+ \backslash \{0\}$. For a DTS $\mathcal{Q} = (V, E, \delta, \{P_i\}_{i \in I})$ and every $v \in V$, we have the following properties.*

Proof.

(1) $f_{\frac{a}{x}}(f_{\frac{a}{x}}(r)) = r$ for every $r \in \mathbb{R}^+_\infty$

(2) $[\![\neg(\varphi \wedge \psi)]\!]^{\mathcal{Q}}(v) = f_{\frac{a}{x}}(\min\{[\![\varphi_1]\!]^{\mathcal{Q}}(v), [\![\varphi_2]\!]^{\mathcal{Q}}(v)\})$
$= \max\{f_{\frac{a}{x}}([\![\varphi_1]\!]^{\mathcal{Q}}(v)), f_{\frac{a}{x}}([\![\varphi_2]\!]^{\mathcal{Q}}(v))\} = [\![\neg\varphi \vee \neg\psi]\!]^{\mathcal{Q}}(v)$.

(3) $[\![\neg(\varphi \vee \psi)]\!]^{\mathcal{Q}}(v) = f_{\frac{a}{x}}(\max\{[\![\varphi_1]\!]^{\mathcal{Q}}(v), [\![\varphi_2]\!]^{\mathcal{Q}}(v)\})$
$= \min\{f_{\frac{a}{x}}([\![\varphi_1]\!]^{\mathcal{Q}}(v)), f_{\frac{a}{x}}([\![\varphi_2]\!]^{\mathcal{Q}}(v))\} = [\![\neg\varphi \wedge \neg\psi]\!]^{\mathcal{Q}}(v)$.

(4) $[\![\neg\Box\varphi]\!]^{\mathcal{Q}}(v) = f_{\frac{a}{x}}(\inf_{v' \in vE} \frac{1}{\delta(v,v')} \cdot [\![\varphi]\!]^{\mathcal{Q}}(v'))$
$= \sup_{v' \in vE} \delta(v, v') \cdot f_{\frac{a}{x}}([\![\varphi]\!]^{\mathcal{Q}}(v')) = [\![\Diamond\neg\varphi]\!]^{\mathcal{Q}}(v)$.

(5) $[\![\neg\Diamond\varphi]\!]^{\mathcal{Q}}(v) = f_{\frac{a}{x}}(\sup_{v' \in vE} \delta(v, v') \cdot [\![\varphi]\!]^{\mathcal{Q}}(v'))$
$= \inf_{v' \in vE} \frac{1}{\delta(v,v')} \cdot f_{\frac{a}{x}}([\![\varphi]\!]^{\mathcal{Q}}(v')) = [\![\Box\neg\varphi]\!]^{\mathcal{Q}}(v)$.

(6) $[\![\neg d \cdot \varphi]\!]^{\mathcal{Q}} = f_{\frac{a}{x}}(d \cdot [\![\varphi]\!]^{\mathcal{Q}}(v))$
$= \frac{1}{d} \cdot f_{\frac{a}{x}}([\![\varphi]\!]^{\mathcal{Q}}(v)) = [\![\frac{1}{d} \cdot \neg\varphi]\!]^{\mathcal{Q}}$.

(7) $[\![\neg\mu X.\varphi]\!]^{\mathcal{Q}} = [\![\nu X.\neg\varphi[X/\neg X]]\!]^{\mathcal{Q}}$.
We will show this case by induction over the stages of the fixed-point evaluation as in Theorem 2.6. Let $[\![\mu X.\varphi]\!]^{\mathcal{Q}} = \lim_n g_n$, and $[\![\nu X.\neg\varphi[X/\neg X]]\!]^{\mathcal{Q}} = \lim_n h_n$. The base case $f_{\frac{a}{x}}(g_0) = h_0$, where $g_0 = 0$ and $h_0 = \infty$ as previously defined, holds by definition of $f_{\frac{a}{x}}$. Assume that $f_{\frac{a}{x}}(g_\alpha) = h_\alpha$ for stage α.

The induction step

$$\begin{aligned} f_{\frac{a}{x}}(g_{\alpha+1}) &= f_{\frac{a}{x}}(\llbracket\varphi\rrbracket^{\mathcal{Q}}_{\mathfrak{I}[X\leftarrow g_\alpha]}) \\ &= \llbracket\neg\varphi\rrbracket^{\mathcal{Q}}_{\mathfrak{I}[X\leftarrow g_\alpha]} \\ &= \llbracket\neg\varphi\rrbracket^{\mathcal{Q}}_{\mathfrak{I}[X\leftarrow f_{\frac{a}{x}}(h_\alpha)]} \\ &= \llbracket\neg\varphi[X/\neg X]\rrbracket^{\mathcal{Q}}_{\mathfrak{I}[X\leftarrow h_\alpha]} = h_{\alpha+1} \end{aligned}$$

follows from the induction hypothesis, the limit step follows trivially.

(8) $\llbracket\neg\nu X.\varphi\rrbracket^{\mathcal{Q}} = \llbracket\mu X.\neg\varphi[X/\neg X]\rrbracket^{\mathcal{Q}}$. The proof is analogous to (7) above. □

Hence, we can add an inductive rule for negation to the definition of dQμ. Moreover, we show that $f_{\frac{a}{x}}$ are the only negation operators for dQμ with the required properties. For this purpose we use the following technical lemma.

Lemma 4.7. *Let $g : \mathbb{R} \to \mathbb{R}$ and $c \in \mathbb{R}$ be such that:*

(1) $g(x+y) = g(x) + g(y) + c$,

(2) $g(g(x)) = x$,

(3) $x < y \implies g(y) < g(x)$.

Then $g(x) = -x - c$.

Proof. First, we establish the following equalities:

(i) $g(0) = -c$, because $g(0+0) = g(0) + g(0) + c$ by (1).

(ii) $g(-x) = -g(x) - 2c$ as $-c = g(0) = g(x + (-x)) = g(x) + g(-x) + c$.

(iii) $g(c) = -2c$ as $g(-c) = 0$ by (2) and (i), and $g(c) = -g(-c) - 2c$ by (ii).

Let us now compare $x + c$ with $-g(x)$ for arbitrary x.

First, if $x + c > -g(x)$, then

$$\begin{aligned} g(x+c) &< g(-g(x)) && \text{by (3)} \\ g(x) + g(c) + c &< -g(g(x)) - 2c && \text{by (1) and (ii)} \\ g(x) - 2c + c &< -x - 2c && \text{by (iii) and (2),} \end{aligned}$$

so $g(x) < -x - c$, which contradicts the assumption that $x + c > -g(x)$.

The case that $x + c < -g(x)$ is treated analogously and also leads to a contradiction. Hence, $-g(x) = x + c$ and therefore also $g(x) = -x - c$ which concludes our proof. $\square$

Note that we prove that $f_{\frac{a}{x}}$ are the only negation operators even for non-discounted transition systems. Observe how each $f_{\frac{a}{x}}$ operates on discounts, thus the function β is $\beta(d) = \frac{1}{d}$. This motivates our definition of the semantics of dQμ, in particular it explains the $\frac{1}{\delta(v,v')}$ factor for $[\![\Box\varphi]\!]^{\mathcal{Q}}$ in Definition 4.3.

Theorem 4.8. *$f_{\frac{a}{x}}$ for $a \in \mathbb{R}^+\setminus\{0\}$ are the only negation operators for* dQμ*, even for non-discounted transition systems.*

Proof. According to property (4) in Definition 4.5, we require $f_\neg(d \cdot x) = \beta(d) \cdot f_\neg(x)$ for some β. If we take $x = 1$, we get $\beta(d) = \frac{f_\neg(d)}{f_\neg(1)}$. Let $a = \frac{1}{f_\neg(1)}$, so $f_\neg(d \cdot x) = f_\neg(d) \cdot f_\neg(x) \cdot a$. Now let $g(x) = \ln f_\neg(e^x)$. From our considerations above, we get

$$g(x + y) = \ln f_\neg(e^{(x+y)}) = \ln f_\neg(e^x \cdot e^y) = g(x) + g(y) + \ln(a).$$

By property (1) we require that $f_\neg(f_\neg(x)) = x$. By definition of g we have $f_\neg(x) = e^{g(\ln(x))}$ which implies that $g(g(\ln(x))) = \ln(x)$. As ln is a function onto $\mathbb{R}^+$, we have $g(g(x)) = x$, and as both ln and exp are monotone, g satisfies conditions $(1) - (3)$ of Lemma 4.7 and thus $g(x) = -x - a$. Thus, $f_\neg(x) = \frac{a}{x}$ and $\beta(d) = \frac{1}{d}$. $\square$

The canonical choice for negation in dQμ is $f_{\frac{1}{x}}$. The dualities between $\wedge$ and $\vee$, $\Diamond$ and $\Box$, and between least and greatest fixed points imply that dQμ has a negation normal form: every formula can be translated into one in which negation is applied only to atoms.

4.3 Discounted Quantitative Parity Games

Discounted quantitative parity games are an extension of quantitative parity games, in so far as they allow discounts on edges. This makes the calculation of payoffs for finite plays slightly more complicated and has some consequences for the properties these games enjoy, e.g. they do in general not admit memory-bounded ε-optimal strategies for a given ε.

We show that discounted quantitative parity games can be used for model checking the discounted μ-calculus. Discounted parity games can also be seen as a more compact representation of infinite quantitative parity games which we discuss in detail later and which enables us to use our previous results to prove the correctness of the model-checking theorem also in the discounted case.

Definition 4.9. A *discounted quantitative parity game* (DPG) is a tuple

$$\mathcal{G} = (V, V_0, V_1, E, \delta, \lambda, \Omega),$$

where V is the disjoint union of V_0 and V_1, i.e. positions belong to either Player 0 or 1. The transition relation $E \subseteq V \times V$ describes possible moves in the game and $\delta : V \times V \to \mathbb{R}^+$ maps every move to a positive real value representing the discount factor. The payoff function $\lambda : \{v \in V : vE = \emptyset\} \to \mathbb{R}^+_\infty$ assigns values to all terminal positions and the priority function $\Omega : V \to \{0, \ldots, n\}$ assigns a priority to every position.

The game is played in the same way as a quantitative parity game. Again, we have two players, Player 0 wants to maximise the outcome and Player 1 wants to minimise it. The main difference to quantitative parity games is that we have to take the discounts into account when calculating the outcome of a finite play.

Formally, the outcome $p(\pi)$ of a finite play $\pi = v_0 \ldots v_k$ is computed by multiplying all discount factors seen throughout the play with the value of the final node,

$$p(v_0 v_1 \ldots v_k) = \delta(v_0, v_1) \cdot \delta(v_1, v_2) \cdot \ldots \cdot \delta(v_{k-1}, v_k) \cdot \lambda(v_k).$$

The outcome of an infinite play again depends only on the lowest priority seen infinitely often. We assign the value 0 to every infinite play where the lowest priority seen infinitely often is odd, and ∞ to those where it is even. The definitions of strategies and determinacy are as before, see Section 3.2.

Classical parity games can also be embedded into quantitative parity games by mapping winning to payoff ∞ and losing to payoff 0 and playing on qualitative arenas (the payoff function only assigns the extremal values 0 or ∞ to terminal nodes). Note that there is no need to consider the discount function δ in the qualitative case as the payoff cannot be changed by discounting.

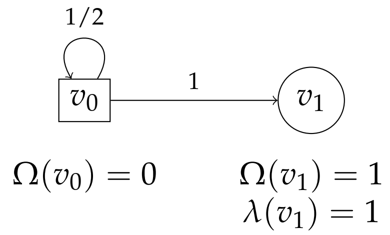


Figure 4.2: No positional or strictly optimal strategy for Player 1

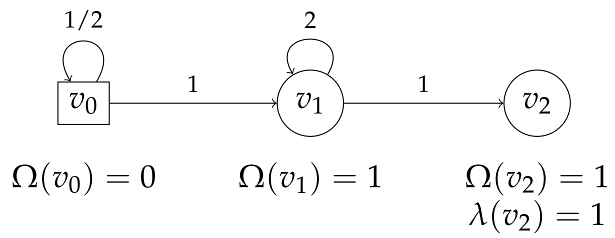


Figure 4.3: No bounded-memory strategy for Player 0 (not even ε-optimal strategy for fixed ε)

As mentioned before, qualitative parity games enjoy the nice property of *positional determinacy* [14, 35, 43]. Non-discounted parity games also admit ε-optimal positional strategies for a fixed ε, but already no optimal strategies, as we have shown in Section 3.1.

Unfortunately, in the case of discounted quantitative parity games, we do not retain this property. There are already simple quantitative games where no player has a positional ε-optimal strategy (for fixed ε). Even worse, in the following examples we present a game where there is no bounded-memory strategy for Player 0, not even an ε-optimal strategy for a fixed ε

Example 4.10. Figure 4.2 shows a simple discounted quantitative parity game. Although it is finite and finitely-branching, we can already see the same phenomenon as in Example 3.3, namely that no optimal strategy exists for Player 1. Again, by convention, we depict positions of Player 0 by circles and the ones of Player 1 by rectangles. The value of this game is 0, as Player 1 can force the value to be arbitrarily small by looping in the first position. He cannot loop indefinitely though, as the priority $\Omega(v_0) = 0$ is even and thus an infinite play would give him the worst possible outcome. However, for every ε he has an

ε-optimal strategy. Obviously, none of these strategies is positional.

In Figure 4.3, we show an extension of the above game, by basically duplicating Player 1's situation (he wants to loop as long as possible, but cannot loop infinitely often) for Player 0. The value of this game is ∞, as Player 0 can enforce arbitrarily high values by looping in v_1, but there is no optimal strategy. Even if one fixes an approximation ε of the game value, Player 0 needs infinite memory to reach this approximation. Please recall that we defined ε-close to ∞ as bigger or equal to $\frac{1}{\varepsilon}$. Consequently, for Player 0 to reach such an outcome in a play, when it is her turn, she first needs to loop in v_1 as long as Player 1 looped in the v_1 to make up for the discounts accumulated so far and get the accumulated discount back to 1. Afterwards, she has to do as many additional loops in v_1 as she needs to get her desired outcome, before she can move to the terminal position v_2.

Crash Games

Crash games, introduced by Gawlitza and Seidl in [20], are an equivalent definition of discounted quantitative parity games where the discounts and payoffs are restricted to the integers instead of the reals. Let $\mathbb{Z}_\infty := \mathbb{Z} \cup \{-\infty, \infty\}$ and $\mathbb{Z}_\infty^+ := \mathbb{Z}_0^+ \cup \{\infty\}$. We give the definition of crash games rephrased in our terminology.

Definition 4.11 ([20])**.** A *crash game* is a tuple

$$\mathcal{G} = (V, V_0, V_1, E, \delta, \Omega),$$

where V is finite and the disjoint union of V_0 and V_1, i.e. positions of Player 0 and 1, and the designated sink position (0), which is the only terminal position (and does not belong to either of the players). The discount function $\delta : V \times V \to \mathbb{Z}$ maps every move to an integer value. The priority function $\Omega : V \to \{0, \ldots, n\}$ assigns a priority to each position.

Please note that Gawlitza and Seidl use an additive version, meaning that the outcome of a finite play is the sum over all discounts seen throughout a play. Formally, the outcome $p(\pi)$ of a finite play $\pi = v_0 \ldots v_k, (0)$ (which has to end in (0)) is computed as

$$p(v_0 v_1 \ldots v_k) = \delta(v_0, v_1) + \delta(v_1, v_2) + \ldots + \delta(v_{k-1}, v_k) + \delta(v_k, (0)).$$

The outcome of an infinite play is determined by the lowest priority occurring infinitely often as for quantitative parity games.

Gawlitza and Seidl solve crash games by reducing them to hierarchical equation systems and then using a variant of strategy improvement, a technique introduced in [42] to solve classical parity games.

Theorem 4.12 ([20])**.** *Let $\mathcal{G} = (V, V_0, V_1, E, \delta, \Omega)$ be a crash game. The values* $\mathrm{val}\mathcal{G}(v)$ *can be computed by a strategy improvement algorithm in time $\mathcal{O}(d \cdot |V|^3 \cdot |\mathcal{G}| \cdot \Pi(|\mathcal{G}|))$, where $|\mathcal{G}| = |V| + |E|$ and d denotes the maximal priority of a position occurring in $\mathcal{G}$.*

$\Pi(m)$ for $m \in \mathbb{N}$ is the maximal number of updates of the strategy improvement algorithm for a game with m positions of Player 0. The trivial upper bound for this number is 2^m. However, the authors claim that in practical implementations this number remains small.

As stated before, crash games are just another representation of finite discounted quantitative parity games over integers. We show how to transform a discounted quantitative parity game over integers into a crash game. First, we have to take into account that in crash games the payoff is calculated as the sum of discounts, whereas in DPGs it is calculated as the product. Also, discounts and payoffs in crash games range over the domain $\mathbb{Z}_\infty := \mathbb{Z} \cup \{-\infty, \infty\}$, but for DPGs we have to restrict this to $\mathbb{Z}^+_\infty := \mathbb{Z}^+_0 \cup \{\infty\}$. This allows us to replace all the discounts on edges by their logarithm. As there is only one terminal node and no payoff function λ in crash games, we also add a designated sink position. Then, we add edges from all former terminal nodes to this new state. These edges get as discount the logarithm of the value of λ.

Formally, for a discounted quantitative parity game $\mathcal{G}$ over $\mathbb{Z}_\infty$,

$$\mathcal{G} = (V, V_0, V_1, E, \delta, \lambda, \Omega) \text{ over } \mathbb{Z}^+_\infty,$$

i.e. $\lambda : \{v \in V \mid vE = \emptyset\} \to \mathbb{Z}^+$ and $\delta : E \to \mathbb{Z}^+$, let

$$\mathcal{G}^c = (V \cup (0), V_0, V_1, E^c, \delta^c, \Omega)$$

be the corresponding crash game where $E^c = E \cup \{(v, (0)) \mid vE = \emptyset\}$ and

$$\delta(v, w) = \begin{cases} \ln \lambda(v, w) & \text{if } w = (0), \\ \ln \delta(v, w) & \text{else.} \end{cases}$$

For a finite play $\pi = v_0, \ldots, v_k, (0)$ in the crash game, we have the payoff

$$\mathrm{p}(\pi) = \delta^c(v_0, v_1) + \ldots + \delta^c(v_k, (0)) = \ln(\delta(v_0, v_1)) + \ldots + \ln(\lambda(v_k)).$$

The value of the corresponding play in $\mathcal{G}$ is calculated using the exponential function.

$$e^{\mathrm{p}(\pi)} = e^{\ln(\delta(v_0,v_1))+\ldots+\ln(\lambda(v_k))} = e^{\ln((\delta(v_0,v_1)\cdot\ldots\cdot\lambda(v_k))} = \delta(v_0,v_1)\cdot\ldots\cdot\lambda(v_k),$$

and thus is equal to the payoff of the corresponding play in the QPG $\mathcal{G}$. For infinite plays, as we have not changed the priority function Ω, we have the same priorities in corresponding plays. Hence, we only have to adjust the payoff according to the slightly different domains, ∞ stays the same (the minimal priority occurring infinitely often in the play is even), but we have to map $-\infty$ in the crash game to 0 in the DPG. The unique correspondence between plays can be extended to strategies, and thus we can use the algorithm to solve crash games also for discounted quantitative parity games over integers.

Corollary 4.13. *Let $\mathcal{G} = (V, V_0, V_1, E, \delta, \lambda, \Omega)$ be a finite discounted quantitative parity game over $\mathbb{Z}_\infty$. The values $\mathrm{val}\mathcal{G}(v)$ can be computed by a strategy improvement algorithm in time $\mathcal{O}(d \cdot |V|^3 \cdot |\mathcal{G}| \cdot \Pi(|\mathcal{G}|))$ where $|\mathcal{G}| = |V| + |E|$ and d denotes the maximal priority of a position occurring in $\mathcal{G}$.*

Please note that also for discounted quantitative parity games over $\mathbb{R}_\infty$, we can easily go back and forth between the version which uses addition and the one that uses multiplication using a logarithm function. This also translates to the logic. For the discounted version of Qμ with addition, the negation operator is thus $f_\neg(x) = -x$.

4.4 Model-Checking Games for Discounted Qμ

As before, we want to show that the value of a formula at a node can be described by the value of the corresponding position in a model-checking game. The model-checking games for dQμ are constructed in a similar way as the ones for Qμ before. The main difference is that we need to add rules to deal with the discounts and the resulting games are discounted parity games.

We briefly repeat the definition of the model-checking game and then state the model-checking theorem for dQμ and discuss how the correctness follows from our previous result.

Definition 4.14. For a discounted quantitative transition system $\mathcal{Q} = (S, T, \delta_S, P_i)$ and a closed dQμ-formula φ in negation normal form, the discounted quantitative parity game

$$\mathrm{MC}[\mathcal{Q}, \varphi] = (V, V_0, V_1, E, \delta, \lambda, \Omega),$$

which we call the *model-checking game* for $\mathcal{Q}$ and φ, is constructed in the following way, similar to Definition 3.5.

Positions. The positions of the game are pairs (ψ, s), where ψ is a subformula of φ, and $s \in S$ is a state of the DTS $\mathcal{Q}$, and the two special positions (0) and (∞). Positions (ψ, s) where the top operator of ψ is $\Box, \wedge$, or ν belong to Player 1 and all other positions belong to Player 0.

Moves. Positions of the form $(|P_i - c|, s)$, $(\neg(|P_i - c|), s)$, (0), and (∞) are terminal positions. From positions of the form $(\psi \wedge \vartheta, s)$, or $(\psi \vee \vartheta, s)$, one can move to (ψ, s) or to (ϑ, s). Positions of the form $(\Diamond\psi, s)$ have either a single successor (0), in case s is a terminal state in $\mathcal{Q}$, or one successor (ψ, s') for every $s' \in sT$. Analogously, positions of the form $(\Box\psi, s)$ have a single successor (∞), if $sT = \emptyset$, or one successor (ψ, s') for every $s' \in sT$ otherwise. Positions of the form $(d \cdot \psi,\ s)$ have a unique successor (ψ, s). Fixed-point positions $(\mu X.\psi, s)$, resp. $(\nu X.\psi, s)$ have a single successor (ψ, s). Whenever one encounters a position where the fixed-point variable stands alone, i.e. (X, s'), the play goes back to the corresponding definition, namely (ψ, s').

Discounts. The discount of an edge is d for transitions from positions $(d \cdot \psi, s)$, it is $\delta_S(s,\ s')$ for transitions from $(\Diamond\psi, s)$ to (ψ, s'), it is $1/\delta_S(s, s')$ for transitions from $(\Box\psi, s)$ to (ψ, s'), and 1 for all outgoing transitions from other positions.

Payoffs. The payoff function λ assigns $|[\![P_i]\!](s) - c|$ or $-|[\![P_i]\!](s) - c|$ to positions $(|P_i - c|, s)$ or $(\neg(|P_i - c|), s)$, ∞ to position (∞), and 0 to position (0).

Priorities. The priority function Ω is defined as in the classical case using the alternation level of the fixed-point variables, see Definition 3.5.

Please note that the main difference to the construction in the non-discounted setting is that we replace the special position $(-\infty)$ by (0) and we are adding the rule for the case that $\varphi = d \cdot \psi$. Payoffs are computed exactly as for discounted quantitative parity games.

Also note that infinite plays will have the outcomes 0 (smallest priority occurring infinitely often is odd) or ∞. As before, we have two players: Verifier wants to maximise the outcome and Falsifier wants to minimise it.

Example 4.15. In Figure 4.4 we depict a model-checking game for $\varphi = \mu X.(P \vee 2 \cdot \Diamond X)$ on the DTS $\mathcal{Q}$ from Example 4.4. The nodes are labelled with the corresponding subformulae of φ, and a state of $\mathcal{Q}$. As there is only one fixed-point variable and no alternation, we only need one priority. Since it is referring to a

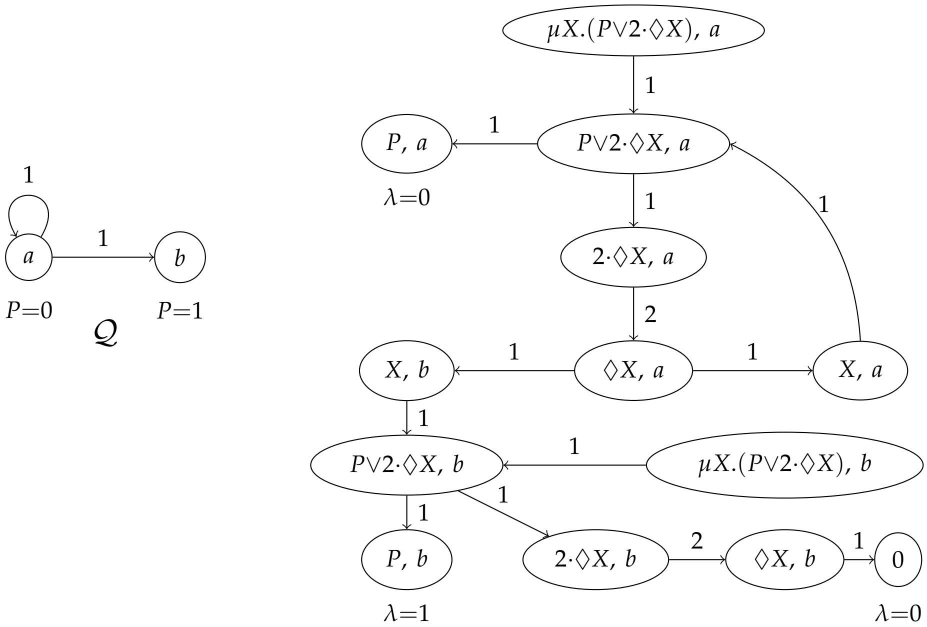


Figure 4.4: DTS $\mathcal{Q}$ and model-checking game for $\mu X.(P \vee 2 \cdot \Diamond X)$ and $\mathcal{Q}$

least fixed point, the (adjusted) priority will be 1 and assigned not only to the fixed-point position but also to all other nodes.

As the only priority is odd, Player 0 has to avoid infinite plays. This is a discounted version of the game in Example 3.8. In the non-discounted game there was no incentive for Player 0 to stay in the game, her best strategy was to move to the terminal node that gives her the payoff 1 immediately. Now the situation has changed. In the discounted game, every loop she makes through the cycle of positions $(P \vee 2 \cdot \Diamond X, a), (2 \cdot \Diamond X, a), (\Diamond X, a), (X, a)$ will double her payoff. So she can reach arbitrarily high outcomes, which means the value of the game starting from $(\mu X.(P \vee 2 \cdot \Diamond X), a)$ is ∞. However, she has no optimal strategy meaning she cannot get this outcome in an actual play.

Now we are ready to state the discounted version of the model-checking theorem.

Theorem 4.16. *For a formula φ in* dQμ*, a discounted quantitative transition system $\mathcal{Q}$, and $v \in \mathcal{Q}$, the game* MC$[\mathcal{Q}, \varphi]$ *is determined and*

$$\mathrm{valMC}[\mathcal{Q}, \varphi](\varphi, v) = [\![\varphi]\!]^{\mathcal{Q}}(v).$$

To prove this theorem, we need to first show that discounted quantitative parity games are determined.

Proposition 4.17. *For a discounted quantitative parity game $\mathcal{G} = (V, E, \lambda, \Omega)$, for all $v \in V$,*

$$\sup_{\sigma \in \Gamma_0} \inf_{\rho \in \Gamma_1} p(\pi_{\sigma,\rho}(v)) = \inf_{\rho \in \Gamma_1} \sup_{\sigma \in \Gamma_0} p(\pi_{\sigma,\rho}(v)) = \mathrm{val}\mathcal{G}(v).$$

Proof. We use the fact that we proved the theorem for non-discounted games of any cardinality. We observe that for any discounted quantitative parity game one can construct an infinite quantitative parity game with the same value. To encode the discounts seen so far in a play in a non-discounted game, we have to enrich the game structure. In the new game, positions are pairs of nodes and a number which represents the current accumulated discounts. This way, when we reach a terminal position, the accumulated discounts up to this point can be determined from the position itself.

Formally, for a DPG

$$\mathcal{G} = (V, V_0, V_1, E, \delta, \lambda, \Omega), \text{ let}$$

$$\mathcal{G}^\delta = (V \times \mathbb{R}_\infty, V_0 \times \mathbb{R}_\infty, V_1 \times \mathbb{R}_\infty, E^\delta, \lambda^\delta, \Omega^\delta) \text{ where}$$

- $E^\delta = \{((v,d),(v',d')) \mid (v,v') \in E \text{ and } d' = d \cdot \delta(v,v')\}$
- $\lambda^\delta : \{v \in V \mid vE = \emptyset\} \times \mathbb{R}_\infty \to \mathbb{R}_\infty$ where $\lambda^\delta(v,d) = d \cdot \lambda(v)$
- $\Omega^\delta(v,d) = \Omega(v)$.

$\mathcal{G}^\delta$ is an infinite quantitative parity game and as such determined as proved in Theorem 3.26.

For a play $\pi = v_0, \ldots, v_i, \ldots$ in $\mathcal{G}$, we have a unique corresponding play in $\mathcal{G}^\delta$, $\pi^\delta = (v_0, 0), \ldots, (v_i, d_i), \ldots$ where $d_i = \prod_{j=0}^{i-1} \delta(v_j, v_{j+1})$ for $i > 0$. The payoffs coincide for corresponding plays: for finite plays $\pi = v_0, \ldots, v_k$ and $\pi^\delta = (v_0, d_0), \ldots, (v_k, d_k)$, we have

$$\mathrm{p}(\pi) = \Big(\prod_{j=0}^{k-1} \delta(v_j, v_{j+1})\Big) \cdot \lambda(v_k) = d_k \cdot \lambda(v_k) = \lambda^\delta(v_k, d_k) = \mathrm{p}(\pi^\delta)$$

For infinite plays, we assign the same priorities to each play and thus the values coincide as well. The unique correspondence between plays can be extended to strategies and thus, the determinacy of discounted quantitative parity games follows from the determinacy of quantitative parity games. $\square$

Please note, that in [17], we give a direct proof of determinacy for arbitrary discounted quantitative parity games and the correctness of model-checking.

After establishing determinacy for discounted quantitative parity games we prove Theorem 4.16, i.e. the correctness of the model-checking game. The proof is almost identical to the non-discounted case, we just have to cover the additional complications brought on by discounting.

Lemma 4.18. $\mathrm{MC}[\mathcal{Q}, \varphi]$ *is a model-checking game for* $\varphi \in \mathrm{dQML}$.

Proof. We prove this by an induction on the structure of the dQML formula. The cases $\varphi = |P_i| - c$, $\varphi = \neg\varphi$, $\varphi = \psi \wedge \vartheta$ and $\varphi = \psi \vee \vartheta$ are analogous to the cases in the proof of Lemma 4.18 for QML.

Thus, we only prove the cases that differ from the QML evaluation, namely the evaluation of modal operators and the new case $\varphi = d \cdot \varphi'$.

In case that $\varphi = \Box\varphi'$, we have $[\![\varphi]\!](v) = \inf_{w \in vE} \frac{1}{\delta(v,w)} [\![\varphi]\!](w)$. Hence, in the model-checking game, the corresponding position belongs to Player 1 and the value $f(v)$ is computed as $\inf_{w \in vE} \frac{1}{\delta(v,w)} f(w)$, where the next positions are the subformula φ' evaluated at each of the successor nodes w in the original transition system. By induction hypothesis, these values coincide with $[\![\varphi']\!](w)$. The case $\varphi = \Diamond\varphi'$ is analogous, but now the position belongs to Player 0.

If $\varphi = d \cdot \varphi'$, we have $[\![\varphi]\!](v) = d \cdot [\![\varphi']\!](v)$. In the model-checking game this position has only one successor $w = vE$, which corresponds to the evaluation of subformula φ' at state v, and the value is computed as $f(v) = d \cdot f(w)$. Again by induction hypothesis, the value $f(w)$ coincides with $[\![\varphi']\!](v)$. $\square$

The proof for the fixed point case remains exactly the same as in the non-discounted case, so we can use Lemma 3.28 and the consequences. Altogether, this concludes the proof of Theorem 4.16.

4.5 Definability of Game Values

Having model-checking games for the quantitative μ-calculus is just one direction in the relation between games and logic. The other direction concerns the definability of winning regions in a game by formulae in the corresponding logic. For the classical μ-calculus such formulae have been constructed by Walukiewicz and it has been shown that for any parity game with a fixed number of priorities they define the winning region for Player 0, see e.g. [24].

We extend this theorem to the discounted case in the following way. We represent discounted quantitative parity games $(V, V_0, V_1, E, \delta_G, \lambda_G, \Omega_G)$ with priorities $\Omega(V) \in \{0, \dots d-1\}$ by a discounted quantitative transition system $\mathcal{Q}_\mathcal{G} = (V, E, \delta, V_0, V_1, \Lambda, \Omega)$, where $V_i(v) = \infty$ if $v \in V_i$ and $V_i(v) = 0$ otherwise, $\Omega(v) = \Omega_G(v)$ if $vE \neq \emptyset$ and $\Omega(v) = d$ otherwise,

$$\delta(v, w) = \begin{cases} \delta_G(v, w) & \text{when } v \in V_0, \\ \frac{1}{\delta_G(v,w)} & \text{when } v \in V_1, \end{cases}$$

and payoff predicate $\Lambda(v) = \lambda_G(v)$ when $vE = \emptyset$ and $\Lambda(v) = 0$ otherwise. We then build the formula Win_d as

$$\mathrm{Win}_d = \nu X_0.\mu X_1.\nu X_2. \dots \lambda X_{d-1} \bigvee_{j=0}^{d-1} ((V_0 \wedge P_j \wedge \Diamond X_j) \vee (V_1 \wedge P_j \wedge \Box X_j)) \vee \Lambda,$$

where $\lambda = \nu$ if d is odd, and $\lambda = \mu$ otherwise, and $P_i := \neg(\mu X.(2 \cdot X \vee |\Omega - i|))$.

Theorem 4.19. *For every $d \in \mathbb{N}$, the value of any discounted quantitative parity game $\mathcal{G}$ with priorities in $\{0, \dots d-1\}$ coincides with the value of Win_d on the associated transition system $\mathcal{Q}_\mathcal{G}$.*

Proof. Please note that $P_i(v) = \infty$ if $\Omega(v) = i$ and $P_i(v) = 0$ otherwise. The formula Win_d is therefore analogous to the one in the qualitative case and the proof is similar as well. We show that the model-checking game for Win_d, $\mathrm{MC}[\mathcal{Q}_\mathcal{G}, \mathrm{Win}_d]$, coincides with $\mathcal{G}$ modulo stupid moves (moves that would lead to an immediate loss for the current player).

We consider a position $v \in V_i$ in original game with priority $\Omega(v) = k$ and distinguish two cases.

If v is a terminal position, then the corresponding DTS also has a terminal position v, where all predicates $P_0, \dots P_{d-1}$ give a value of 0. In the game $\mathrm{MC}[\mathcal{Q}_\mathcal{G}, \mathrm{Win}_d]$ the play goes to a position (Λ, v) which gives a value of $\lambda_G(v)$ as in the original game.

If v is non-terminal of priority k, then the corresponding DTS $\mathcal{Q}_\mathcal{G}$ has a position v, where $V_i(v) = \infty$ and $P_k(v) = \infty$, and the discounts on the outgoing edges are $\delta_G(v, w)$ if $i = 0$, or $\frac{1}{\delta_G(v,w)}$ if $i = 1$. The game $\mathrm{MC}[\mathcal{Q}_\mathcal{G}, \mathrm{Win}_d]$ then gets to

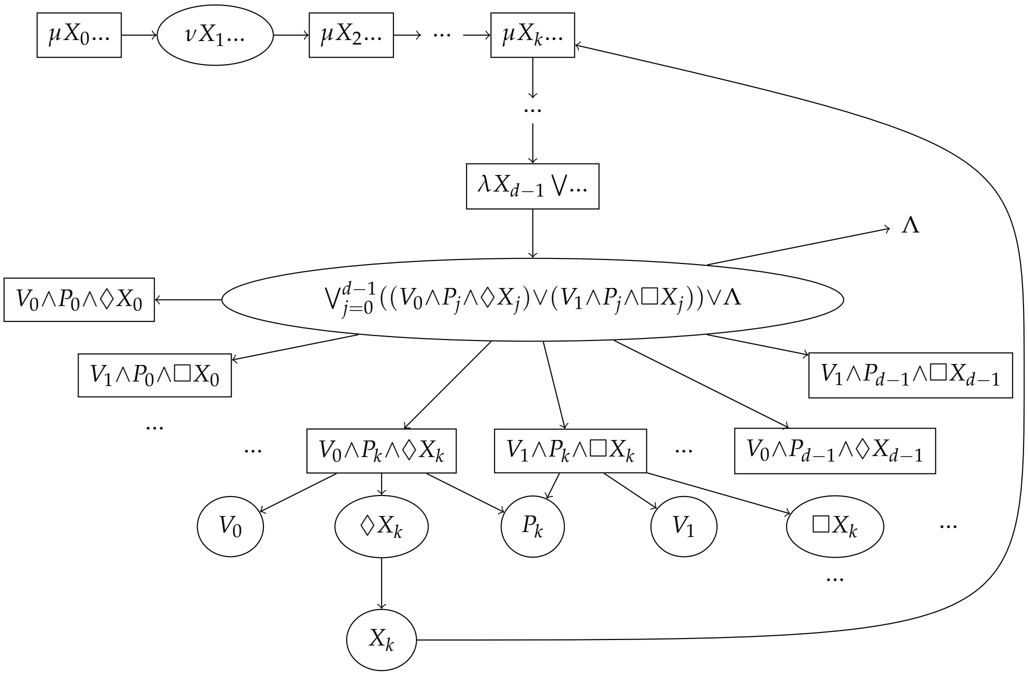


Figure 4.5: Configuration in the model-checking game for Win_d

a position $(\Diamond X_k, v)$ if $i = 0$ or $(\Box X_k, v)$ if $i = 1$, except for the case that one of the players makes an immediately losing move. If the players avoid that, then Player i makes a move that exactly corresponds to a move in the original game from v to some successor w, visits a position (X_k, w) with priority k and the situation repeats for w as depicted in Figure 4.5.

Let us convince you that indeed the game will proceed to position (Λ, v), $(\Diamond X_k, v)$ or $(\Box X_k, v)$, or else some player has made a stupid move. We only consider the case for $v \in V_0$ in the original game. In the positions corresponding to the series of fixed points preceding the subformula

$$\vartheta = \bigvee_{j=0}^{d-1} ((V_0 \wedge P_j \wedge \Diamond X_j) \vee (V_1 \wedge P_j \wedge \Box X_j)) \vee \Lambda,$$

there is no choice for either of the players, so the play proceeds to the position (ϑ, v) and it is Player 0's turn to choose a successor. If v is terminal position, the only reasonable choice is to go to the position (Λ, v), as for all other positions Player 1 can make a move to a terminal position (P_i, v) for $i = 0, \ldots d-1$, which will give a payoff of 0. If v is non-terminal and has priority k in the original game, i.e. $P_j(v) = 0$ for $j \neq k$, and $\Lambda(v) = 0$, then any other move than to a subformula containing the right priority predicate P_k would be pointless. Of course, Player 0 will go to the position corresponding to the disjunct $(V_0 \wedge P_k \wedge \Diamond X_k)$ as we assumed that the position belonged to her in the original game, and therefore $V_0(v) = \infty$. In this position, Player 1 may choose the next position, but again any other move than the one to $(\Diamond X_k, v)$ would be stupid, since all the other predicates will give a value of ∞ by choice of Player 0 in the last move. From position $(\Diamond X_k, v)$, Player 0 can choose a successor w, i.e. lead the game to position (X_k, w) with priority k. The discount $\delta((\Diamond X_k, v), (X_k, w))$ is the same as in the original game by construction of $\mathcal{Q}_\mathcal{G}$ and $\mathrm{MC}[\mathcal{Q}_\mathcal{G}, \mathrm{Win}_d]$.

The other case, $v \in V_1$, is analogous, the only difference is that now the play will go through a position $(\Box X_k, v)$, where Player 1 can choose a successor (X_k, w).

Hence, the two games $\mathcal{G}$ and $\mathrm{MC}[\mathcal{Q}_\mathcal{G}, \mathrm{Win}_d]$ coincide, i.e. all the relevant choices to be made by the players and priorities seen throughout the plays are essentially the same. Therefore, for all $v \in V$,

$$\mathrm{val}\mathcal{G}(v) = \mathrm{val}\mathrm{MC}[\mathcal{Q}_\mathcal{G}, \mathrm{Win}_d](\mathrm{Win}_d, v) = [\![\mathrm{Win}_d]\!]^{\mathcal{Q}_\mathcal{G}}(v).$$

□

Please note that we used discounting in Win_d as a trick to ensure value ∞ if we are at the right priority (encoded by the quantitative predicate) and 0 for all other values of predicates. We cannot do this in the non-discounted case and thus have not established this direction of the model-checking theorem in the non-discounted setting.

5 Qμ on Linear Hybrid Systems

So far, we have investigated the quantitative μ-calculus on simple quantitative transition systems and the slightly more complex discounted quantitative transition systems. In both cases, we have defined appropriate model-checking games and have shown that they correctly describe the value of a quantitative formula. In this chapter, we want to move further towards possible applications and focus on a more complex class of systems. Because of their importance in practice, an obvious candidate is the class of hybrid systems. However, these systems are inherently difficult to handle and almost all interesting questions regarding hybrid systems are undecidable [27]. Thus, we look at a sub class which is still practically relevant, namely the class of initialised linear hybrid systems. We show that an appropriate version of the quantitative μ-calculus, hybrid Qμ, can be model-checked with arbitrary precision on initialised linear hybrid systems.

A hybrid system is, as the name suggests, a hybrid of a discrete transition system and continuous variables which evolve according to a set of differential equations. It is a concept widely used in engineering to model discrete-continuous systems. As the variables in each state have quantitative values, it seems very natural to query hybrid systems in a quantitative way instead of only asking yes-or-no questions. 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.

Because of their importance in practical applications, model-checking techniques have been applied to hybrid systems to verify safety, liveness and other classical temporal properties [1, 26, 27]. However, quantitative testing of hybrid systems has only been done by simulation, and hence lacks the strong guarantees which can be given by model checking. As we have noted before, there has been a strong interest recently in extending classical model-checking techniques and logics to the quantitative setting. Although there has been a rise in introducing quantitative versions of temporal logics, those were mostly evaluated on labelled, timed, or probabilistic transition systems. None of those systems allowed for dynamically changing continuous variables.

We want to apply our logic to a class of hybrid systems but, as we have mentioned before, for general hybrid systems, even simple qualitative verification problems such as reachability are undecidable. 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 [32, 4] and their recent extensions [41] – one of the largest classes of hybrid systems with a decidable temporal logic [27].

We show that the quantitative μ-calculus can be model checked with arbitrary precision on initialised systems. Thus, we present the first model-checking algorithm for a quantitative temporal logic on a class of hybrid systems. Since the quantitative μ-calculus contains LTL, this also properly generalises a previous result on model checking LTL on such systems [26, 27], which is one of the strongest model-checking results for hybrid systems.

As in the previous chapters, we follow the classical approach to model-checking the μ-calculus via parity games. To this end, we define another quantitative notion of parity games that we call interval parity games. Again, we use our previous results on quantitative parity games to show that interval parity games can be used to model check the hybrid quantitative μ-calculus on linear hybrid systems.

We proceed by simplifying the resulting games to flat games, i.e. simple games where all the linear coefficients are 1. These games look very similar to timed games with more complex payoff rules but unfortunately behave differently. Thus, the methods used for solving timed games turned out not to be sufficient for our games and did not easily generalise to the quantitative case. We overcome this problem by working directly with a quantitative equivalence relation, roughly similar to the region graph for timed automata and introducing a new class of (almost) discrete strategies. This way, we are able to reduce the model-checking games to the much simpler class of counter-reset games. Then, we can exploit a recent result on solving counter parity games, which are a generalisation of counter-reset games [2].

The organisation of this chapter follows the reductions needed to model-check a formula φ over a hybrid system $\mathcal{K}$. In Section 5.1, we introduce the necessary notation, the systems, and the logic. Then, we present an appropriate game model in Section 5.2 and show how to construct a model-checking game for the system and the formula. In Section 5.4, we transform interval parity games constructed for arbitrary initialised linear hybrid systems to flat games, where the linear coefficients are always 1. In Section 5.5, we show how the strategies can be discretised and still lead to a good approximation of the original game. Finally, in Section 5.6, we reduce the problem to counter parity games and exploit a recent result to solve them.

5.1 Syntax and Semantics of Hybrid Qμ

As before, we denote the real and rational numbers, and integers extended with both ∞ and $-\infty$ by $\mathbb{R}_\infty$, $\mathbb{Q}_\infty$, and $\mathbb{Z}_\infty$ 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 $\mathbb{R}_\infty$ with endpoints in $\mathbb{Z}_\infty, \mathbb{Q}_\infty$, and $\mathbb{R}_\infty$.

Definition 5.1. A *linear hybrid system over M variables* (LHS),

$$\mathcal{K} = (V, E, \{P_i\}_{i \in J}, \lambda, \kappa),$$

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 $\lambda : E \to \mathcal{P}_{\text{fin}}(\mathcal{L}_M)$ assigns to each transition a finite set of labels. The set $\mathcal{L}_M$ of *transition labels* consists of triples $l = (I, \overline{C}, R)$. The vector $\overline{C} = (C_1, \ldots, C_M)$ (with $C_i \in \mathcal{I}(\mathbb{R}_\infty)$ for $i \in \{1, \ldots, M\}$) 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. 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 $P_i : V \to \mathbb{R}_\infty$ assigns to each location the value of the static quantitative predicate P_i. The function $\kappa : V \to \mathbb{R}^M$ 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 $\mathbb{Q}_\infty$ or $\mathbb{Z}_\infty$ instead of $\mathbb{R}_\infty$, 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 $s = (v, y_1, \ldots, y_M)$ we say that

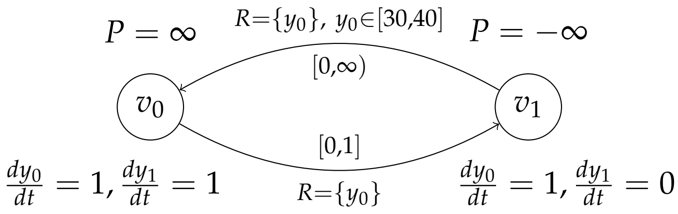


Figure 5.1: Leaking gas burner LHS $\mathcal{L} = (V, E, P, \lambda, \kappa)$ (not initialised)

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 $i = 1, \ldots, M$). We say that a state $s' = (v', y'_1, \ldots, y'_M)$ is a *successor* of s, denoted $s' \in \mathrm{succ}(s)$, if 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 $a_i = \kappa_i(v)$ 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 $s_0, s_1, \ldots$ such that $s_0 = (v_0, 0, \ldots, 0)$ and $s_{i+1} \in \mathrm{succ}(s_i)$ for all i. Given two states s and $s' \in \mathrm{succ}(s)$ and a reset set $R \neq \{1, \ldots, M\}$ 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 5.2. A linear hybrid system $\mathcal{K}$ is *initialised* if for each $(v, w) \in E$ and each variable y_i it holds that if $\kappa_i(v) \neq \kappa_i(w)$ 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 5.3. 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 5.1. 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

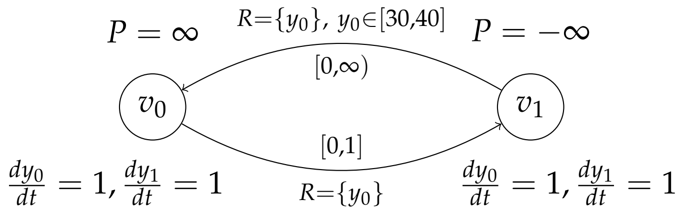


Figure 5.2: Leaking gas burner LHS $\mathcal{L} = (V, E, P, \lambda, \kappa)$ (initialised)

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 5.2, 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.

Now, we present a version of the quantitative μ-calculus suited to be evaluated on linear hybrid systems. It only differs slightly from our definition in Chapter 2, but for convenience we repeat the whole definition.

Definition 5.4. Given a set $\mathcal{X}$ of fixed-point variables X, system variables $\{y_1, \ldots, y_M\}$ and predicates $\{P_i\}_{i \in J}$, the formulae of the *hybrid quantitative μ-calculus (*hQμ*)* are built in the following way:

(1) P_i is a hQμ-formula,

(2) X is a hQμ-formula,

(3) y_i is a hQμ-formula,

(4) if φ is a hQμ-formula, then so is $\neg\varphi$,

(5) if φ, ψ are hQμ-formulae, then so are $(\varphi \wedge \psi)$ and $(\varphi \vee \psi)$,

(6) if φ is a hQμ-formula, then so are $\Box\varphi$ and $\Diamond\varphi$,

(7) if φ is a formula of hQμ, then $\mu X.\varphi$ and $\nu X.\varphi$ are formulae of hQμ given that X occurs only positively (i.e. under an even number of negations) in φ.

Let $\mathcal{F} = \{f : S \to \mathbb{R}_\infty\}$ be the set of functions from the states of a LHS to the reals.

Definition 5.5. Given a linear hybrid system $\mathcal{K} = (V, E, \lambda, \{P_i\}_{i\in J}, \kappa)$ and an interpretation $\mathfrak{I} : \mathcal{X} \to \mathcal{F}$, a hQ$\mu$-formula yields a valuation function $[\![\varphi]\!]^{\mathcal{K}}_{\mathfrak{I}} : S \to \mathbb{R}_\infty$ defined in the following standard way for a state $s = (v^s, y^s_1, \ldots, y^s_M)$.

(1) $[\![P_i]\!]^{\mathcal{K}}_{\mathfrak{I}}(s) = P_i(v^s)$,

(2) $[\![X]\!]^{\mathcal{K}}_{\mathfrak{I}}(s) = \mathfrak{I}(X)(s)$,

(3) $[\![y_i]\!]^{\mathcal{K}}_{\mathfrak{I}}(s) = y^s_i$,

(4) $[\![\neg\varphi]\!]^{\mathcal{K}}_{\mathfrak{I}}(s) = -[\![\varphi]\!]^{\mathcal{K}}_{\mathfrak{I}}(s)$,

(5) $[\![\varphi_1 \wedge \varphi_2]\!]^{\mathcal{K}}_{\mathfrak{I}}(s) = \min\{[\![\varphi_1]\!]^{\mathcal{K}}_{\mathfrak{I}}(s), [\![\varphi_2]\!]^{\mathcal{K}}_{\mathfrak{I}}(s)\}$,
$[\![\varphi_1 \vee \varphi_2]\!]^{\mathcal{K}}_{\mathfrak{I}}(s) = \max\{[\![\varphi_1]\!]^{\mathcal{K}}_{\mathfrak{I}}(s), [\![\varphi_2]\!]^{\mathcal{K}}_{\mathfrak{I}}(s)\}$,

(6) $[\![\Diamond\varphi]\!]^{\mathcal{K}}_{\mathfrak{I}}(s) = \sup_{s'\in\mathrm{succ}(s)}[\![\varphi]\!]^{\mathcal{K}}_{\mathfrak{I}}(s')$,
$[\![\Box\varphi]\!]^{\mathcal{K}}_{\mathfrak{I}}(s) = \inf_{s'\in\mathrm{succ}(s)}[\![\varphi]\!]^{\mathcal{K}}_{\mathfrak{I}}(s')$,

(7) $[\![\mu X.\varphi]\!]^{\mathcal{K}}_{\mathfrak{I}}(s) = \inf\{f \in \mathcal{F} : f = [\![\varphi]\!]^{\mathcal{K}}_{\mathfrak{I}[X\leftarrow f]}\}(s)$,
$[\![\nu X.\varphi]\!]^{\mathcal{K}}_{\mathfrak{I}}(s) = \sup\{f \in \mathcal{F} : f = [\![\varphi]\!]^{\mathcal{K}}_{\mathfrak{I}[X\leftarrow f]}\}(s)$.

Please note that the inclusion of variables does not fundamentally change the semantics of quantitative μ-calculus. Qμ is evaluated on quantitative transition systems. A hQμ 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 remain true. For example, to embed the classical μ-calculus in hQμ one must interpret *true* as $+\infty$ and *false* as $-\infty$.

Example 5.6. 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 $(v_0, \overline{0})$ in our example.

The remainder of this chapter is dedicated to the proof of the following result which states that $[\![\varphi]\!]^{\mathcal{K}}$ can be approximated with arbitrary precision on initialised linear hybrid systems.

Theorem 5.7. *Given an initialised linear hybrid system $\mathcal{K}$, a quantitative μ-calculus formula φ and an integer $n > 0$, it is decidable whether $[\![\varphi]\!]^{\mathcal{K}} = \infty$, $[\![\varphi]\!]^{\mathcal{K}} = -\infty$, or else a number $r \in \mathbb{Q}$ can be computed such that $|[\![\varphi]\!]^{\mathcal{K}} - r| < \frac{1}{n}$.*

In other words, for every ε we can approximate $[\![\varphi]\!]^{\mathcal{K}}$ within ε. We stated the theorem above using n because it makes the representation of ε precise and we 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.

5.2 Interval Parity Games

In this section, we define a variant of quantitative parity games suited for model checking hQμ 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 Chapter 3.

Definition 5.8. An *interval parity game* (IPG) is given by a tuple,

$$\mathcal{G} = (V_0, V_1, E, \lambda, \kappa, \iota, \Omega),$$

and is played on a LHS (V, E, λ, κ) (without predicates). $V = V_0 \,\dot\cup\, V_1$ 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 $\lambda : E \to \mathcal{P}_{\text{fin}}(\mathcal{L}_M)$. The function $\iota : V \to M \times \mathbb{R}_\infty \times \mathbb{R}_\infty$ 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 $\Omega : V \to \{0, \dots, d\}$ 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* $s = (v, \overline{y}) \in V \times \mathbb{R}_\infty^M$ 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 $\mathrm{loc}(s) = v$ and $\mathrm{var}(s) = \overline{y}, \mathrm{var}_i(s) = y_i$ to access the components of a state. For a real number r, we denote by $r \cdot s = (v, r \cdot \mathrm{var}_0(s), \ldots r \cdot \mathrm{var}_M(s))$ and $r + s = (v, r + \mathrm{var}_0(s), \ldots r + \mathrm{var}_M(s))$. We call S_i the state set $\{s = (v, \overline{y}) : v \in V_i\}$ where player i has to move and $S = S_0 \cup S_1$.

How to play. Every play starts at some position $v_0 \in V$ with all variables set to 0, i.e. the starting state is $s_0 = (v_0, 0, \ldots, 0)$. For every state $s = (v, \overline{y}) \in S_i$, player i chooses an allowed successor state $s' \in \mathrm{succ}(s)$ and the play proceeds from s'. If the play reaches a state s such that $\mathrm{succ}(s) = \emptyset$ 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 $\mathrm{p}(s_0, \ldots, s_k)$ of a finite play ending in $s_k = (v, y_1, \ldots, y_M)$ where $\iota(v) = (i, a, b)$ is $\mathrm{p}(s_k) = a \cdot y_i + b$. 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 $\sigma : S^*S_i \to S$ with $\sigma(s_0, \ldots, s_n) \in \mathrm{succ}(s_n)$. A play $\pi = s_0 s_1 \ldots$ is *consistent with a strategy* σ for player i, if $s_{n+1} = \sigma(s_0 \ldots s_n)$ 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.

The notion of determinacy is as before, see Section 3.1. 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 5.9. In Figure 5.3, we show a 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, κ assigns 1 to every location, 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 ∞.

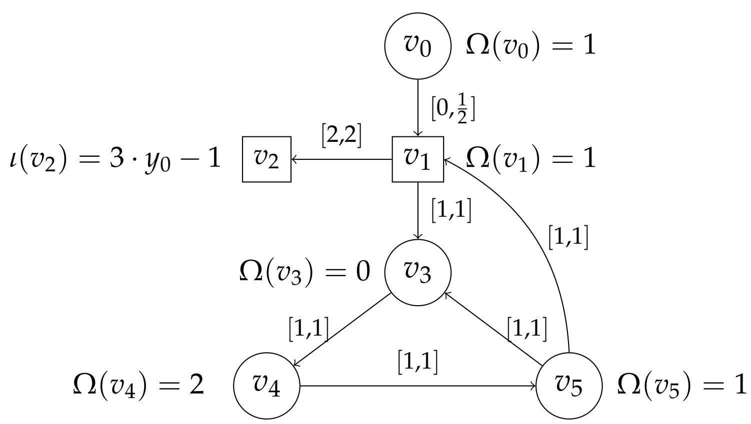


Figure 5.3: Simple interval parity game

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 Definition 3.1 and define, for an IPG with M variables

$$\mathcal{G} = (V_0, V_1, E, \lambda, \kappa, \iota, \Omega),$$

the corresponding infinite quantitative parity game

$$\mathcal{G}^* = (V_0 \times \mathbb{R}^M_\infty, V_1 \times \mathbb{R}^M_\infty, E^*, \lambda^*, \Omega^*)$$

- with $(s, s') \in E^*$ iff s' is a successor of s as above,
- $\Omega^*(v, \overline{z}) = \Omega(v)$ and
- $\lambda^*(v, \overline{z}) = \alpha \cdot z_i + \beta$ 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 particular, it follows from the determinacy of quantitative parity games that also interval parity games are determined.

5.3 Model-Checking Games for Hybrid Qμ

For a linear hybrid system $\mathcal{K}$ and a Qμ-formula φ, we construct an interval parity game MC$[\mathcal{K}, \varphi]$ which is the model-checking game for φ on $\mathcal{K}$.

The full definition of MC$[\mathcal{K}, \varphi]$ closely follows the construction given in Chapter 3.2 and is presented below.

Definition 5.10. For a linear hybrid system $\mathcal{K} = (V, E, \{P_i\}_{i \in J}, \lambda, \kappa)$ and a closed hQμ-formula φ in negation normal form, the interval game

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

which we call the *model-checking game* for $\mathcal{K}$ and φ, is constructed in the following way, similarly to the standard construction of model-checking games for the μ-calculus.

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 $\Box, \wedge$, or ν belong to Player 1 and all other positions belong to Player 0. A state in the game is denoted by $s = (p, \overline{y})$, where $p = (\psi, v)$ is the position and $\overline{y}$ is the variable assignment of the location v in the underlying linear hybrid system $\mathcal{K}$.

Moves. Positions of the form (P_i, v), $(\neg P_i, v)$, (y_i, v), and $(\neg y_i, v)$ are terminal positions. From positions of the form $(\psi \wedge \vartheta, v)$, resp. $(\psi \vee \vartheta, v)$, 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 $(\Box\psi, v)$ 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 $[\![P_i]\!](v)$ to all positions (P_i, v), $[\![\neg P_i]\!](v)$ to positions $(\neg P_i, v)$, $\pm\infty$ to all positions $(\pm\infty)$ and y_i to positions (y_i, v), and $-y_i$ to $(\neg y_i, v)$. 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 positions or ∞ for Player 1's positions. The payoff $\mathrm{p}(\pi)$ 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 as before.

Example 5.11. We continue our example of the leaking gas burner and present in Figure 5.4 the model-checking game for the system depicted in Figure 5.2 and the formula $\mu X.(\Diamond X \vee (y_0 \wedge P))$ from Example 5.6. 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.

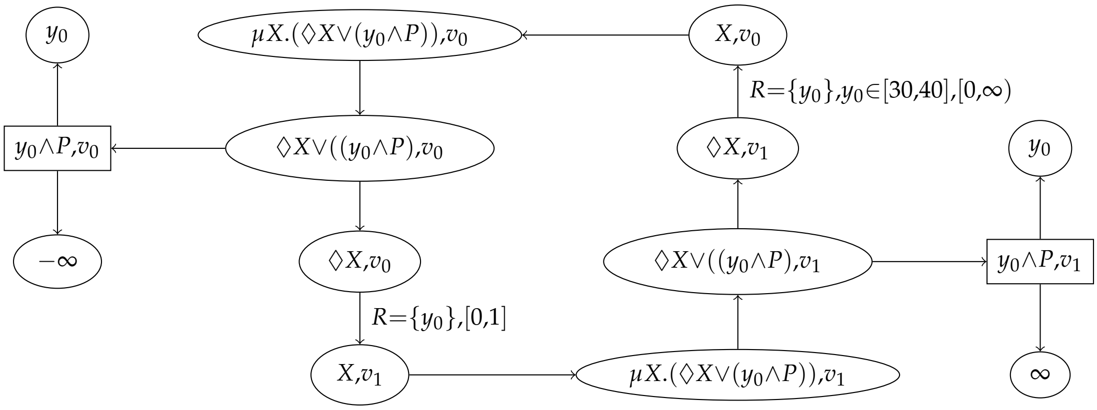


Figure 5.4: Model-checking game for $\mu X.(\Diamond X \vee (y_0 \wedge P))$ on initialised leaking gas burner

We have shown in Section 3.2 that quantitative parity games of any size are determined and that they are model-checking games for Qμ. These results translate to interval parity games and we can conclude the following.

Theorem 5.12. *Every interval parity game is determined and for every formula φ in* hQμ*, linear hybrid system $\mathcal{K}$, and a location v of $\mathcal{K}$, it holds that*

$$\text{valMC}[\mathcal{K},\varphi]((\varphi,v),\overline{0}) = [\![\varphi]\!]^{\mathcal{K}}(v,\overline{0}).$$

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 Qμ-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 $P_{y_i}(v,\overline{a}) = a_i$. 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 3.7, we conclude that for all $v \in \mathcal{K}^*$ it holds $\text{valMC}[\mathcal{K}^*,\varphi^*](\varphi^*,v) = [\![\varphi]\!]^{*\mathcal{K}^*}(v)$, 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 Qμ on the other, it follows that for all $\overline{x}$,

$$\text{valMC}[\mathcal{K},\varphi]((\varphi,v),\overline{x}) = [\![\varphi]\!]^{\mathcal{K}}(v,\overline{x}).$$

□

5.4 Simplifying Interval Parity Games

Before we start simplifying interval parity games, we first note that they are not equivalent to timed systems, although they may look similar. Below, we give a simple example that illustrates the difference between the two. Then, we show how to transform an initialised interval game over $\mathbb{Q}_\infty$ into an easier game over $\mathbb{Z}_\infty$ in which all evolution rates are one.

At first sight, interval parity games look similar to timed games. These 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 5.5 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 $\mathbb{Z}_\infty$. 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.

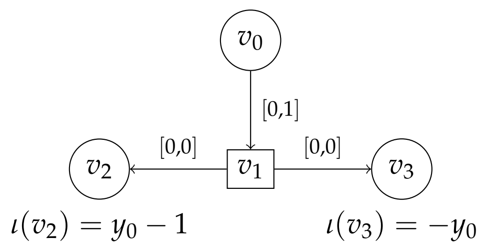


Figure 5.5: Game with integer coefficients and non-integer value

Flattening Initialised Interval Parity 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 5.13. An interval parity game $\mathcal{G} = (V_0, V_1, E, \lambda, \kappa, \iota, \Omega)$ is *flat* if and only if $\kappa_i(v) = 1$ for all $v \in V$ and $i = 1 \dots M$.

Lemma 5.14. *For each initialised interval parity game $\mathcal{G}$, there exists a flat game $\mathcal{G}'$ with the same value.*

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

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

Note that we only change the functions κ, λ and ι. We 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 $\mathrm{p}(\pi) = \mathrm{p}(\pi')$. Before we proceed with the proof, please note 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 $s_0 = (v_0, \overline{0})$ is a state belonging to Player 0 and $\sigma(s_0) = s_1 = (v_1, \overline{x})$ and $s'_0 = (v_0, \overline{0})$, then in $\mathcal{G}'$ we define $\sigma'(s'_0) = s'_1$, where $s'_1 = (v_1, \overline{y}')$, 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 $\mathrm{p}(s_1) = \iota(v_1) = a \cdot y_i + b$ and therefore the payoff is equal to

$$\mathrm{p}(s'_1) = \iota'(v'_1) = a_i \cdot a \cdot \frac{y_i}{a_i} + b.$$

Let $s_0, \ldots, s_k$ and $s'_0, \ldots, s'_k$ be finite histories in $\mathcal{G}$ and $\mathcal{G}'$, such that they visit the same locations and $\mathrm{p}(\pi) = \mathrm{p}(\pi')$. If $s_k = (v_k, \overline{y})$ is a state belonging to Player 0 and $\sigma(s_k) = s_{k+1} = (v_{k+1}, \overline{y})$ and $s'_k = (v_k, \overline{z})$, then we define $\sigma'(s'_k) = s'_{k+1}$ in $\mathcal{G}'$, where $s'_{k+1} = (v_k, \overline{w})$, 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 $\mathrm{p}(s_k) = \iota(v_k) = a \cdot y_i + b$ and therefore, the payoff is equal to

$$\mathrm{p}(s'_{k+1}) = \iota'(v'_{k+1}) = a_i \cdot a \cdot w_i + b = a_i \cdot a \cdot \frac{y_i}{a_i} + b.$$

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. □

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

Multiplying Interval Parity Games

Definition 5.15. 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 $(I', \overline{C'}, R) \in \lambda'((v, w))$ iff $(I, \overline{C}, R) \in \lambda((v, w))$ 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 equal to the values of $\mathcal{G}$ multiplied by q.

Lemma 5.16. *For every IPG $\mathcal{G}$ over $\mathbb{Q}_\infty$ and $q \in \mathbb{Q}, q \neq 0$ it holds in all states s that $q \cdot \mathrm{val}\mathcal{G}(s) = \mathrm{val}\, q \cdot \mathcal{G}(q \cdot s)$.*

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

$$q \cdot \mathrm{p}_{\mathcal{G}}(\pi(\sigma, \rho, s)) = q \cdot \mathrm{p}_{\mathcal{G}}(s_0 s_1 \dots s_k) = q \cdot (a \cdot y_i + b)$$

where $\iota(\mathrm{loc}(s_k)) = (a, i, b)$ which is equal to

$$\mathrm{p}_{q \cdot \mathcal{G}}(\pi(q \cdot \sigma, q \cdot \rho, q \cdot s)) = \mathrm{p}_{q \cdot \mathcal{G}}(q \cdot s_0 \dots q \cdot s_k) = a \cdot (q \cdot y_i) + q \cdot b$$

for all finite plays π. Therefore, we know that

$$\inf_{\rho} q \cdot \mathrm{p}(\pi(\sigma, \rho, s)) = \inf_{q \cdot \rho} \mathrm{p}(\pi(q \cdot \sigma, q \cdot \rho, q \cdot s))$$

and the same holds for the supremum and thus, we get the desired result. □

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

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

5.5 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 $\mathbb{Z}_\infty$. 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 for 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 5.18. We say that two states s and t in an IPG are equivalent, $s \sim t$, if they are in the same location, $\mathrm{loc}(s) = \mathrm{loc}(t)$, and for all $i, j \in \{1, \ldots, K\}$:

– $[\mathrm{var}_i(s)] = [\mathrm{var}_i(t)]$, and

– if $\{\mathrm{var}_i(s)\} \leq \{\mathrm{var}_j(s)\}$ then $\{\mathrm{var}_i(t)\} \leq \{\mathrm{var}_j(t)\}$.

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 5.19. *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 $R = \{1, \ldots, K\}$ 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 $R \neq \{1, \ldots, K\}$ 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., $\{\mathrm{var}_i(t)\} \leq \{\mathrm{var}_j(t)\}$ for all $j \notin R$. To construct t', we must choose w' with $[w'] = [w]$ which makes $\mathrm{var}_i(s' + w')$ the one with smallest fractional part.

Case 1: $\{\mathrm{var}_i(t)\} \geq \{w\}$.
In this case, for all non-reset variables j, it holds that $\{\mathrm{var}_j(t)\} \geq \{w\}$, intuitively meaning that no variable "jumped" above an integer due to $\{w\}$ (illustrated in Figure 5.6). 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

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

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

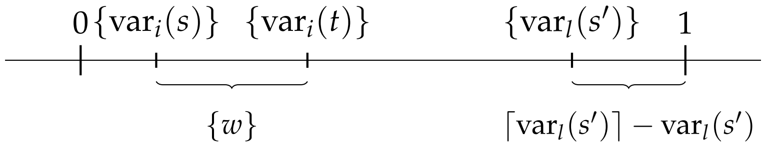


Figure 5.6: Lemma 5.19, case 1

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 $\{w\}$ (illustrated in Figure 5.7). Let l be the variable with maximum fractional part in s' (and thus also in s). Let

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

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 $[w'] = [w]$ and both the order of fractional parts and integer bounds in t' are the same as in t, since

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

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

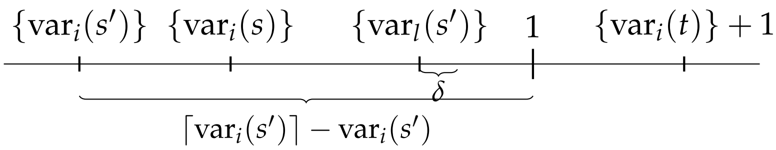


Figure 5.7: Lemma 5.19, case 2

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

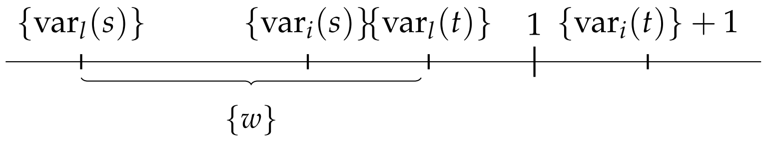


Figure 5.8: Lemma 5.19, case 3 for s

one which did not "jump" above an integer due to $\{w\}$. The variable with the next greatest fractional part in s (and by $\sim$ also in s') is $\mathrm{var}_i(s)$, as depicted in Figure 5.8.

To transfer the move to s', consider these two variables in s' as depicted in Figure 5.9 and let $\delta = \{\mathrm{var}_i(s')\} - \{\mathrm{var}_l(s')\}$.

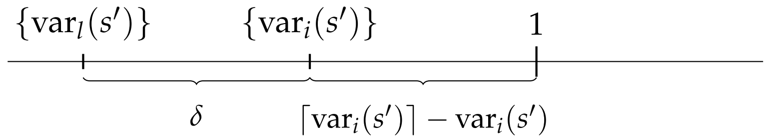


Figure 5.9: Lemma 5.19, case 3 for s'

We set

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

Again $[w'] = [w]$ 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 $t \sim t'$. $\square$

Knowing that we can shift a single move and preserve $\sim$-equivalence, we proceed to show that for IPGs over $\mathbb{Z}_\infty$, fully general strategies are not necessary. In fact, we can restrict ourselves to discrete strategies and, using this, reduce the games to discrete ones. 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

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

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 5.10. 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.

Figure 5.10: Move where standard distance is necessarily increased

To give a more suitable notion of distance for a state, let us, for $r \in \mathbb{R}$, define

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

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 5.11.

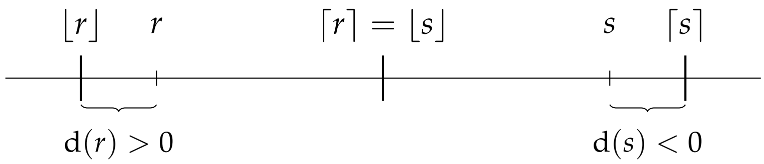


Figure 5.11: Notation for distances between real numbers and integers

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

$$|\mathrm{d}(a+b)| \leq |\mathrm{d}(a)| + |\mathrm{d}(b)|.$$

Also, we observe that

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

For a state s, we use the abbreviation $\mathrm{d}_i(s) = \mathrm{d}(\mathrm{var}_i(s))$. We denote by $\mathrm{d}_l(s) = \min_{i=1\ldots k}\{\mathrm{d}_i(s)\}$ and $\mathrm{d}_r(s) = \max_{i=1\ldots k}\{\mathrm{d}_i(s)\}$ the smallest and greatest of all values $\mathrm{d}_i(s)$, and additionally we define the total distance as follows

$$\mathrm{d}^*(s) = \begin{cases} |\mathrm{d}_l(s)| & \text{if } \mathrm{d}_i(s) \leq 0 \text{ for all } i \in \{1, \ldots, k\}, \\ \mathrm{d}_r(s) & \text{if } \mathrm{d}_i(s) \geq 0 \text{ for all } i \in \{1, \ldots, k\}, \\ |\mathrm{d}_l(s)| + \mathrm{d}_r(s) & \text{otherwise.} \end{cases}$$

This is illustrated in Figure 5.12, 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 greatest one greater than $\frac{1}{2}$ and y_2 has the greatest fractional part (less than $\frac{1}{2}$).

First, we prove that we can always correct a strategy that makes one step which is not ε-discrete. By doing so, we 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.

Lemma 5.20. *Let s be a state with $\mathrm{d}^*(s) \leq \frac{1}{4}$ and t be a successor of s, where (s, t) is allowed by $l = (I, \overline{C}, R)$. Then, for every $0 \leq \varepsilon < \mathrm{d}^*(s)$, there exists a successor t'_+ of s such that*

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

Proof. We assume that $\mathrm{d}^*(t) > \mathrm{d}^*(s) + \varepsilon$, 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 $\mathrm{d}^*(t)$.

Case 1: $\mathrm{d}^*(t) = |\mathrm{d}_l(t)|$.
We correct w in the following way: $w' = w + \mathrm{c} - \varepsilon$, where $\mathrm{c} = \min\{|\mathrm{d}_r(t)|, |d(w)|\}$ if $\mathrm{d}(w) < 0$ and $\mathrm{c} = |\mathrm{d}_r(t)|$ otherwise.

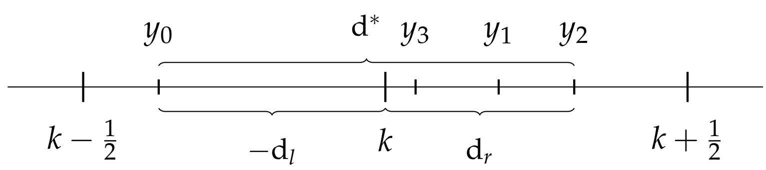


Figure 5.12: Maximal, minimal and total distance for a state

First, we have to show that $[w'] \in [w]$ and therefore $w' \in I$. Since $\mathrm{d}_l(t) = \mathrm{d}_i(t) = \mathrm{var}_i(t)$ for one i, we can conclude from

$$|\mathrm{d}(\mathrm{var}_i(s) + w)| \leq |\mathrm{d}(\mathrm{var}_i(s))| + |\mathrm{d}(w)|$$

that $|\mathrm{d}(w)| > \varepsilon$ and therefore $w' \geq w$, hence $w' \geq \lfloor w \rfloor$. Furthermore, $w' \leq \lceil w \rceil$. Otherwise, if $\mathrm{d}(w) < 0$ then

$$w' = w + \mathrm{c} - \varepsilon > \lceil w \rceil = w + |\mathrm{d}(w)|.$$

This is a contradiction, since by definition $\mathrm{c} \leq |\mathrm{d}(w)|$.

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

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 $\mathrm{var}_i(t'_+) \geq \lfloor \mathrm{var}_i(t) \rfloor$ for all $i \notin R$. We now have to show that also $\mathrm{var}_i(t'_+) \leq \lceil \mathrm{var}_i(t) \rceil$. Let j be the index of the variable which is the closest to the integers (in this case), i.e. j, such that $\mathrm{d}(\mathrm{var}_j(t)) = \mathrm{d}_r(t)$.

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

Also, we have to show: $\mathrm{d}^*(t'_+) \leq \mathrm{d}^*(s) + \varepsilon$. We know that

$$|\mathrm{d}_l(t)| - |\mathrm{d}_r(t)| \leq \mathrm{d}^*(s) \text{ and } \mathrm{d}^*(t'_+) = |\mathrm{d}_l(t'_+)| = |\mathrm{d}(\mathrm{var}_j(t'_+))|$$

for one j and $\mathrm{var}_j(t'_+) = \mathrm{var}_j(s) + w + \mathrm{c} - \varepsilon$. Hence,

$$\mathrm{d}(\mathrm{var}_j(t'_+)) = \mathrm{d}_l(t) + \mathrm{c} - \varepsilon, \text{ since } |\mathrm{d}_l(t) + \mathrm{c} - \varepsilon| \leq \frac{1}{2}.$$

We can conclude that $\mathrm{d}_l(t'_+) = \mathrm{d}(\mathrm{var}_j(t'_+)) \leq \mathrm{d}^*(s) + \varepsilon$.

Case 2: $\mathrm{d}^*(t) = |\mathrm{d}_r(t)|$.

Subcase 1: $\mathrm{d}(w) > 0$*:*

We correct w in the following way:

$$w' = w + (1 - \mathrm{c}) - \varepsilon, \text{ where } \mathrm{c} = \max\{|\mathrm{d}_l(t)|, |d(w)|\}.$$

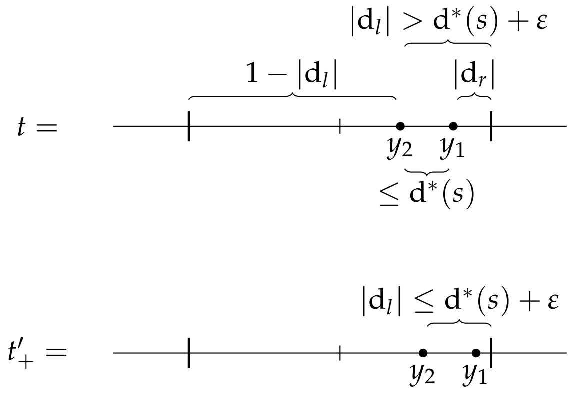


Figure 5.13: Lemma 5.20, case 1

First, we have to show that $[w'] \in [w]$ and therefore $w' \in I$.
Since $\mathrm{d}_r(t) = \mathrm{d}_i(t) = \mathrm{var}_i(t)$ for one i, we can conclude from

$$|\mathrm{d}(\mathrm{var}_i(s) + w)| \leq |\mathrm{d}(\mathrm{var}_i(s))| + |\mathrm{d}(w)|$$

that $|\mathrm{d}(w)| > \varepsilon$ and therefore $w' \geq w$, hence $w' \geq \lfloor w \rfloor$. Furthermore, $w' \leq \lceil w \rceil$. Otherwise, since $\mathrm{d}(w) > 0$ and we assume that

$$w' = w + (1 - \mathrm{c}) - \varepsilon > \lceil w \rceil = w + (1 - |\mathrm{d}(w)|).$$

This is a contradiction, since by definition $\mathrm{c} \geq |\mathrm{d}(w)|$.

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 $\mathrm{var}_i(t'_+) \geq \lfloor \mathrm{var}_i(t) \rfloor$ for all $i \notin R$. We now have to show that also $\mathrm{var}_i(t'_+) \leq \lceil \mathrm{var}_i(t) \rceil$. Let j be the index of the variable which is the closest to the integers (in this case), i.e. j, such that $\mathrm{d}(\mathrm{var}_j(t)) = \mathrm{d}_l(t)$.

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

Also, we have to show: $\mathrm{d}^*(t'_+) \leq \mathrm{d}^*(s) + \varepsilon$. We know that

$\mathrm{d}_r(t) - \mathrm{d}_l(t) \leq \mathrm{d}^*(s)$ and $\mathrm{d}^*(t'_+) = |\mathrm{d}_r(t'_+)| = |\mathrm{d}(\mathrm{var}_j(t'_+))|$ for one j and

$$\mathrm{var}_j(t'_+) = \mathrm{var}_j(s) + w + (1 - \mathrm{c}) - \varepsilon.$$

Hence,

$$\mathrm{d}(\mathrm{var}_j(t'_+)) = \mathrm{d}_r(t) + (1 - \mathrm{c}) + \varepsilon - 1 = \mathrm{d}_r(t) - \mathrm{c} + \varepsilon.$$

We can conclude that

$$\mathrm{d}_r(t'_+) = \mathrm{d}(\mathrm{var}_j(t'_+)) \leq \mathrm{d}^*(s) + \varepsilon.$$

by definition of c.

Subcase 2: $\mathrm{d}(w) < 0$:
In this case, from $\mathrm{d}^*(s) < \frac{1}{4}$ and $\mathrm{d}^*(t) = \mathrm{d}_r(t)$ it follows that $\mathrm{d}(var_i(s)) < 0$ for all i. Thus, we set $w' = w + \lceil w \rceil - \varepsilon$ and the lemma holds.

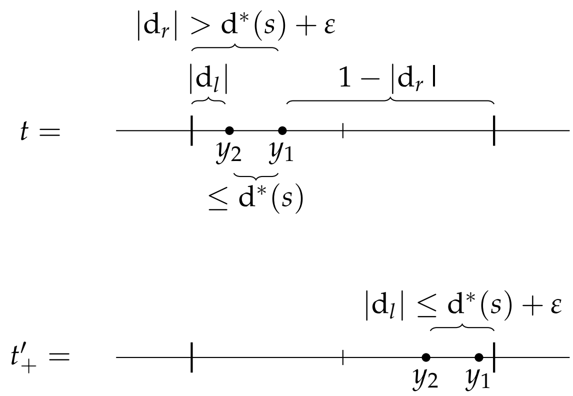


Figure 5.14: Lemma 5.20, case 2

Case 3: $\mathrm{d}^*(t) = \mathrm{d}_r(t) + |\mathrm{d}_l(t)|$.
We correct w in the following way:

$$w' = w + \mathrm{c} - \frac{\varepsilon}{2} \text{ where } \mathrm{c} = \min\{|\mathrm{d}_l(t)|, |d(w)|\}.$$

First, we have to show that $[w'] \in [w]$ and therefore $w' \in I$. Since $\mathrm{d}_r(t) = \mathrm{d}_i(t) = \mathrm{var}_i(t)$ for one i and $\mathrm{d}_l(t) = \mathrm{d}_j(t) = \mathrm{var}_j(t)$ for one j, we can conclude from

$$\begin{aligned}
&|\mathrm{d}(\mathrm{var}_i(s) + w)| \leq |\mathrm{d}(\mathrm{var}_i(s))| + |\mathrm{d}(w)| \\
\text{and } &|\mathrm{d}(\mathrm{var}_j(s) + w)| \leq |\mathrm{d}(\mathrm{var}_j(s))| + |\mathrm{d}(w)| \\
\text{and } &|\mathrm{d}(\mathrm{var}_j(s) + w)| + |\mathrm{d}(\mathrm{var}_i(s) + w)| \\
&\leq |\mathrm{d}(\mathrm{var}_i(s))| + |\mathrm{d}(w)| + |\mathrm{d}(\mathrm{var}_j(s))| + |\mathrm{d}(w)| \leq \mathrm{d}^*(s) + 2|\mathrm{d}(w)| \\
\text{and } &|\mathrm{d}(\mathrm{var}_j(s) + w)| + |\mathrm{d}(\mathrm{var}_i(s) + w)| > \mathrm{d}^*(s) + \varepsilon
\end{aligned}$$

therefore $|\mathrm{d}(w)| > \frac{\varepsilon}{2}$. Hence, $w' \geq \lfloor w \rfloor$. Furthermore, $w' \leq \lceil w \rceil$, otherwise if $\mathrm{d}(w) < 0$ then assume

$$w' = w + \mathrm{c} - \frac{\varepsilon}{2} > \lceil w \rceil = w + |\mathrm{d}(w)|.$$

Then $\mathrm{c} - \frac{\varepsilon}{2} > |\mathrm{d}(w)|$. Contradiction. Otherwise, if $\mathrm{d}(w) > 0$, then $w' \leq \lceil w \rceil$, since by definition $\mathrm{c} \leq \frac{1}{2}$.

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 $\mathrm{var}_i(t'_+) \geq \lfloor \mathrm{var}_i(t) \rfloor$ for all $i \notin R$. We now have to show that also $\mathrm{var}_i(t'_+) \leq \lceil \mathrm{var}_i(t) \rceil$. Let j be the index of the variable with $\mathrm{d}(\mathrm{var}_j(t)) = \mathrm{d}_l(t)$.

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

Thus we have to show: $\mathrm{d}^*(t'_+) \leq \mathrm{d}^*(s) + \varepsilon$. We know that

$$|\mathrm{d}_r(t) - (1 + \mathrm{d}_l(t))| \leq \mathrm{d}^*(s) \text{ and } \mathrm{d}^*(t'_+) = |\mathrm{d}_l(t'_+)| = |\mathrm{d}(\mathrm{var}_i(t'_+))|$$

for j such that $\mathrm{d}(\mathrm{var}_j(t)) = \mathrm{d}_r(t)$. Also,

$$\mathrm{var}_j(t'_+) = \mathrm{var}_j(s) + w + \mathrm{c} - \frac{\varepsilon}{2}.$$

We can conclude that $\mathrm{d}^*(t'_+) \leq \mathrm{d}^*(s) + \frac{\varepsilon}{2}$.

□

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

Definition 5.21. We call a strategy σ *ε-discrete* if for every $s_{n+1} = \sigma(s_0, \ldots, s_n)$ it holds that if $\mathrm{d}^*(s_n) \leq \varepsilon$ then $\mathrm{d}^*(s_{n+1}) \leq \mathrm{d}^*(s_n) + \frac{\varepsilon}{2^{n+1}}$, and if for each i $s'_i \sim s_i$, then $\sigma(s_0, \ldots, s_n) \sim \sigma(s'_0, \ldots, s'_n)$.

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

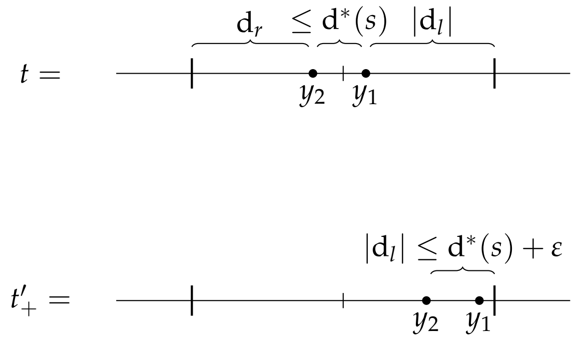


Figure 5.15: Lemma 5.20, case 3

Example 5.22. To see that decreasing ε in each step is sometimes crucial, consider the game with one variable depicted in Figure 5.16. 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.

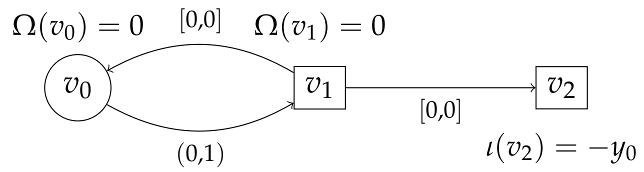


Figure 5.16: Flat IPG where Player 0 has to decrease the value in every step

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

Lemma 5.23. *Let s be a state and t a successor of s, where (s, t) is allowed by l. Let s' be a state with $\mathrm{d}^*(s') \leq \frac{1}{4}$, 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*
- *$\mathrm{d}^*(t') \leq \mathrm{d}^*(s') + \varepsilon$.*

Proof. Since $s \sim s'$ and $t \in \mathrm{succ}(s)$ is allowed by l, we know, by Lemma 5.19, that there exists a state $t' \in \mathrm{succ}(s')$ allowed by the same label l, such that $t' \sim t$. We also know from Lemma 5.20 that, for every choice of ε, there exists $t_+ \in \mathrm{succ}(s')$ such that $\mathrm{d}^*(t_+) \leq \mathrm{d}^*(s') + \varepsilon$ and $t' \sim t_+$. Since $t' \sim t$, this also means that $t_+ \sim t$, hence t_+ fulfils the requirements above. □

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

Lemma 5.24. *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 $\mathrm{d}^*(s_0) < \frac{\varepsilon}{2}$, if $\pi(\sigma, \rho_d, s_0) = s_0, s_1, \ldots$ and $\pi(\sigma_d, \rho_d, s_0) = s'_0, s'_1, \ldots$, 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 5.23, there is a $s'_1 \sim s_1$ with $\mathrm{d}^*(s'_1) \leq \mathrm{d}^*(s_0) + \frac{\varepsilon}{4}$, and we set $\sigma_d(s_0) = s'_1$.

Let $h = s_0, \ldots, s_k$ and $h' = s'_0, \ldots, s'_k$ 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 $\mathrm{d}^*(s_k) \leq \varepsilon(1 - \frac{1}{2^{k+1}})$. If $\sigma(s_0 \ldots s_k) = s_{k+1} \in \mathrm{succ}(s_k)$, then, by Lemma 5.23, there also exists a state $s'_{k+1} \in \mathrm{succ}(s'_k)$ such that $s'_{k+1} \sim s_{k+1}$ and $\mathrm{d}^*(s'_{k+1}) \leq \mathrm{d}^*(s_k) + \frac{\varepsilon}{2}$. Thus, we set $\sigma_d(s'_0 \ldots s'_k)$ to s'_{k+1}. For all other histories $h'' = s''_0, \ldots, s''_k$ 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 5.19, and we can pick a discrete one if $\mathrm{d}^*(s''_k) < \varepsilon$ by Lemma 5.23.

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

Proposition 5.25. *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*

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

Proof. Case 1: assume that

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

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

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

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

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

From Lemma 5.24, we know that there is a discrete strategy $\rho_{\inf_d} \in \Delta_1$ which is a discrete version of $\rho_{\inf}$ against σ_d. From the above, it follows that

$$\mathrm{p}(\pi(\sigma_d, \rho_{\inf_d}, s)) - \mathrm{p}(\pi(\sigma_d, \rho_{\inf}, s)) > m.$$

This is a contradiction, since we know from Lemma 5.24 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

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

By Theorem 3.7 every interval parity game is determined, thus

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

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

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

Therefore we can rewrite the assumption of this case as

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

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

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

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

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

From Lemma 5.24, we know, that there again is a discrete strategy $\sigma_{\sup_d} \in \Delta_0$ which is a discrete version of $\sigma_{\sup}$ against ρ_d. From the above, it follows that

$$\mathrm{p}(\pi(\sigma_{\sup}, \rho_d, s)) - \mathrm{p}(\pi(\sigma_{\sup_d}, \rho_d, s)) > m,$$

which again contradicts that all states in these two plays are equivalent. □

5.6 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 $\mathbb{Z}_\infty$ can be reduced to solving a counter parity game. We then solve these games using an algorithm from [2] for counter parity games, a generalisation of our games.

By Proposition 5.25 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 $n \cdot m \cdot \mathcal{G}$. If the players only use discrete strategies, the chosen values remain close to integers (possibly being up to ε greater or smaller). It can be stored in the state whether the value is greater, equal or smaller than an integer as well as whether the value of a variable is smaller or greater 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 5.26. 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, reset or left intact.

Example 5.27. In Figure 5.6, we depict a simple counter-reset game. As usual, circles represent positions of Player 0 and rectangles 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 $-\infty$.

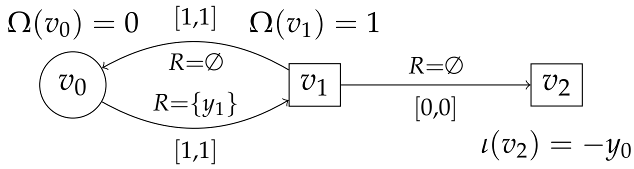


Figure 5.17: Simple counter-reset game

Lemma 5.28. *Let $\mathcal{G}$ be an IPG over $\mathbb{Z}_\infty$ 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:*

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

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 5.16, in all states s holds

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

Moreover, by Proposition 5.25 applied to $\mathcal{G}''$

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

and therefore,

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

We show how to construct the counter-reset game $\mathcal{G}'$ with value equal to,

$$\sup_{\sigma\in\Delta_0}\inf_{\rho\in\Delta_1} \mathrm{p}(\pi_{G''}(\sigma,\rho,s)),$$

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 $n \le l \le n+k$ 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 $j_k \ge i_k$ (interval was downwards-closed), and

– if $l = k$ 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,

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

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 a special case of a class of counter parity games which were recently studied by Berwanger, Leßenich and Kaiser in [2]. Counter parity games are more general than counter-reset games, they allow the counters to be updated by arbitrary affine transformations. The authors give an algorithm to solve such games, improving our previous decidability result [18]. Interestingly, the determinacy of counter parity games is a consequence of our determinacy result for quantitative parity games, Proposition 3.26.

Theorem 5.29 ([2])**.** *For any finite counter parity game $\mathcal{G}$ and initial vertex v, the value* $\mathrm{val}\mathcal{G}(v)$ *can be computed in* 6EXPTIME*. When the number of counters is fixed, the value can be computed in* 4EXPTIME*.*

Corollary 5.30. *For any finite counter-reset game $\mathcal{G}$ with a starting state s where all counters are integers, the value* $\mathrm{val}\mathcal{G}(s)$ *can be computed in* 6EXPTIME*. With fixed number of counters, the value can be computed in* 4EXPTIME*.*

We conclude by completing the proof of our main Theorem 5.7. We first observe that, by Theorem 5.12, evaluating a hQμ-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 5.14 and then into one over $\mathbb{Z}_\infty$ by Corollary 5.17. By Lemma 5.28, the value of such a game can be approximated with arbitrary precision by counter-reset games, which we can solve by Corollary 5.30.

Altogether, 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 5.28 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 5.30). Although there is room for improvement regarding the complexity, our result lays a foundation for using quantitative temporal logics in the verification of hybrid systems.

Bibliography

[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 (1995), 3–34. (Cited on page 87.)

[2] Dietmar Berwanger, Łukasz Kaiser and Simon Lessenich. Solving counter parity games. In *Proceedings of the 37th International Symposium on Mathematical Foundations of Computer Science, MFCS 2012* (2012), vol. 7464 of LNCS, Springer. (Cited on pages 88, 115, 117 and 118.)

[3] Alexis-Julien Bouquet, Olivier Serre and Igor Walukiewicz. Pushdown games with unboundedness and regular conditions. In *Proceedings of the IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2003* (2003), vol. 2914 of LNCS, Springer, pp. 88–99. (Cited on page 11.)

[4] Patricia Bouyer, Thomas Brihaye and Fabrice Chevalier. Weighted o-minimal hybrid systems are more decidable than weighted timed automata! In *Proceedings of the International Symposium on Logical Foundations of Computer Science, LFCS 2007* (2007), vol. 4514 of LNCS, Springer, pp. 69–83. (Cited on page 88.)

[5] Julian Bradfield and Colin Stirling. Modal mu-calculi. In *Handbook of Modal Logic* (2007), Elsevier, pp. 721–756. (Cited on pages 3, 4 and 13.)

[6] Julian C. Bradfield and Colin Stirling. Modal logics and mu-calculi: an introduction. In *Handbook of Process Algebra*. Elsevier, 2001, pp. 293–332. (Cited on page 1.)

[7] Luca de Alfaro. Quantitative verification and control via the mu-calculus. In *Proceedings of the 14th international Conference on Concurrency Theory, Concur 2003* (2003), vol. 2761 of LNCS, Springer, pp. 102–126. (Cited on page 5.)

[8] Luca de Alfaro, Marco Faella, Thomas A. Henzinger, Rupak Majumdar and Mariëlle Stoelinga. Model checking discounted temporal properties. *Theoretical Computer Science 345*, 1 (2005), 139–170. (Cited on page 5.)

[9] Luca de Alfaro, Marco Faella and Mariëlle Stoelinga. Linear and branching metrics for quantitative transition systems. In *Proceedings of the 31st International Colloquium on Automata, Languages and Programming, ICALP 2004* (2004), vol. 3142 of LNCS, Springer, pp. 97–109. (Cited on pages 5, 7 and 67.)

[10] Luca de Alfaro, Marco Faella and Mariëlle Stoelinga. Linear and branching system metrics. *IEEE Transactions on Software Engineering 35*, 2 (2009), 258–273. (Cited on pages 5, 7, 21, 22, 31, 32, 67 and 70.)

[11] Luca de Alfaro, Thomas A. Henzinger and Rupak Majumdar. Discounting the future in systems theory. In *Proceedings of the 30st International Colloquium on Automata, Languages and Programming, ICALP 2003* (2003), vol. 2719 of LNCS, Springer, pp. 1022–1037. (Cited on page 5.)

[12] Luca de Alfaro and Rupak Majumdar. Quantitative solution of omega-regular games. *Journal of Computer and Systems Sciences 68*, 2 (2004), 374–397. (Cited on page 5.)

[13] Yuxin Deng, Tom Chothia, Catuscia Palamidessi and Jun Pang. Metrics for action-labelled quantitative transition systems. *Electronic Notes in Theoretical Computer Science 153*, 2 (2006), 79–96. (Cited on page 5.)

[14] E. Allen Emerson and Charanjit S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science, FOCS 1991* (1991), IEEE, pp. 368–377. (Cited on pages 37, 40 and 76.)

[15] E. Allen Emerson, Charanjit S. Jutla and A. Prasad Sistla. On model checking for the μ-calculus and its fragments. *Theoretical Computer Science 258*, 1–2 (2001), 491–522. (Cited on pages 37 and 43.)

[16] Diana Fischer, Erich Grädel and Łukasz Kaiser. Model checking games for the quantitative mu-calculus. In *Proceedings of the 25th Annual Symposium on Theoretical Aspects of Computer Science, Bordeaux, France,*

STACS 2008 (2008), vol. 1 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, pp. 301–312. (Cited on pages 8 and 67.)

[17] Diana Fischer, Erich Grädel and Łukasz Kaiser. Model checking games for the quantitative μ-calculus. *Theory of Computing Systems 47*, 3 (2010), 696–719. (Cited on pages 8, 67 and 83.)

[18] Diana Fischer and Łukasz 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)* (2011), vol. 6756 of LNCS, Springer, pp. 404–415. (Cited on pages 10 and 118.)

[19] Diana Fischer and Łukasz Kaiser. Model checking the quantitative mu-calculus on linear hybrid systems. *Logical Methods in Computer Science 8*, 3 (2012). (Cited on page 10.)

[20] Thomas Gawlitza and Helmut Seidl. Computing game values for crash games. In *Proceedings of the 5th International Symposium on Automated Technology for Verification and Analysis, ATVA 2007* (2007), vol. 4762 of LNCS, Springer, pp. 177–191. (Cited on pages 68, 77 and 78.)

[21] Hugo Gimbert. Parity and exploration games on infinite graphs. In *Proceedings of the Annual Conference of the European Association for Computer Science Logic, CSL 2004* (2004), vol. 3210 of LNCS, Springer, pp. 56–70. (Cited on page 11.)

[22] Hugo Gimbert and Wieslaw Zielonka. Perfect information stochastic priority games. In *Proceedings of ICALP'07* (2007), vol. 4596 of LNCS, Springer, pp. 850–861. (Cited on page 5.)

[23] Valentin Goranko and Martin Otto. *Handbook of Modal Logic*. Elsevier, 2006, ch. Model Theory of Modal Logic, pp. 255–325. (Cited on pages 3, 4 and 23.)

[24] Erich Grädel. Finite model theory and descriptive complexity. In *Finite Model Theory and Its Applications*. Springer, 2007, pp. 125–230. (Cited on pages 3, 17, 37, 43, 50 and 84.)

[25] Matthew Hennessy and Robin Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM 32*, 1 (Jan. 1985), 137–161. (Cited on pages 3 and 36.)

[26] Thomas A. Henzinger, Benjamin Horowitz and Rupak Majumdar. Rectangular hybrid games. In *Proceedings of the 10th international Conference on Concurrency Theory, Concur 1999* (1999), vol. 1664 of LNCS, Springer, pp. 320–335. (Cited on pages 10, 87 and 88.)

[27] Thomas A. Henzinger, Peter W. Kopke, Anuj Puri and Pravin Varaiya. What's decidable about hybrid automata? In *Proceedings of the 27th Annual ACM Symposium on Theory of Computing, STOC 1995* (1995), ACM, pp. 373–382. (Cited on pages 9, 10, 87 and 88.)

[28] Wilfrid Hodges. *Model Theory*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1993. (Cited on page 23.)

[29] Marcin Jurdziński. Small progress measures for solving parity games. In *Proceedings of the 17th Annual Symposium on Theoretical Aspects of Computer Science, STACS 2000* (2000), vol. 1770 of LNCS, Springer-Verlag, pp. 290–301. (Cited on page 3.)

[30] Łukasz Kaiser and Simon Lessenich. A Counting Logic for Structure Transition Systems. In *Proceedings of the 21st EACSL Annual Conferences on Computer Science Logic, CSL 2012* (2012), vol. 16 of LIPIcs, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pp. 366–380. (Cited on page 10.)

[31] Dexter Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science 27* (1983), 333–354. (Cited on pages 1 and 13.)

[32] Gerardo Lafferriere, George J. Pappas and Shankar Sastry. O-minimal hybrid systems. *Mathematics of Control, Signals, and Systems 13* (2000), 1–21. (Cited on page 88.)

[33] Annabelle McIver and Carroll Morgan. Results on the quantitative μ-calculus qMμ. *ACM Transactions on Computational Logic 8*, 1 (2007). (Cited on page 5.)

[34] Matteo Mio. On the equivalence of game and denotational semantics for the probabilistic mu-calculus. *Logical Methods in Computer Science 8*, 2 (2012). (Cited on page 5.)

[35] Andrzej Włodzimierz Mostowski. Games with forbidden positions. Tech. Rep. 78, Instytut Matematyki, Uniwersytet Gdański, Poland, 1991. (Cited on pages 37, 40 and 76.)

[36] David Michael Ritchie Park. Concurrency and automata on infinite sequences. In *Proceedings of the 5th GI-Conference on Theoretical Computer Science (TCS'81)* (1981), P. Deussen, Ed., vol. 104 of Lecture Notes in Computer Science, Springer-Verlag, pp. 167–183. (Cited on page 3.)

[37] Wolfgang Thomas. On the Ehrenfeucht-Fraïssé game in theoretical computer science. In *Proceedings of Theory and Practice of Software Development, TAPSOFT'93* (1993), Lecture Notes in Computer Science, Springer, pp. 559–568. (Cited on page 23.)

[38] Johan van Benthem. Correspondence theory. In *Handbook of Philosophical logic*, D. M. Gabbay and F. Guenthner, Eds., vol. 4. Kluwer Academic Publishers, 1983, pp. 162–167. (Cited on page 3.)

[39] Johan van Benthem. Exploring logical dynamics. *Journal of Logic, Language and Information 9*, 2 (2000), 261–263. (Cited on page 3.)

[40] Franck van Breugel. A behavioural pseudometric for metric labelled transition systems. In *Proceedings of the 16th international Conference on Concurrency Theory, Concur 2005* (2005), vol. 3653 of Lecture Notes in Computer Science, Springer, pp. 141–155. (Cited on pages 5, 21, 22, 31 and 32.)

[41] Vladimeros Vladimerou, Pavithra Prabhakar, Mahesh Viswanathan and Geir E. Dullerud. Stormed hybrid systems. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming, ICALP'08 (2)* (2008), vol. 5126 of LNCS, Springer, pp. 136–147. (Cited on page 88.)

[42] Jens Vöge and Marcin Jurdziński. A discrete strategy improvement algorithm for solving parity games. In *Proceedings of the 12th International Conference on Computer Aided Verification, CAV 2000* (2000), vol. 1855 of LNCS, Springer-Verlag, pp. 202–215. (Cited on page 78.)

[43] Wiesław Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theoretical Computer Science 200*, 1–2 (1998), 135–183. (Cited on pages 37, 40 and 76.)

Index