Regret and Partial Observability in Quantitative Games

Guillermo A. Pérez

Université Libre de Bruxelles,
Département d'Informatique
September 2016

This dissertation is presented in fulfillment of the requirements for the degree of *Docteur en Sciences*.

Abstract

Two-player zero-sum games of infinite duration and their quantitative versions are used in verification to model the interaction between a controller (Eve) and its environment (Adam). The question usually addressed is that of the existence (and computability) of a strategy for Eve that can maximize her payoff against any strategy of Adam: a *winning strategy*. It is often assumed that Eve always knows the exact state of the game, that is, she has full observation.

In this dissertation, we are interested in two variations of quantitative games. First, we study a different kind of strategy for Eve. More specifically, we consider strategies that minimize her *regret*: the difference between her actual payoff and the payoff she could have achieved if she had known the strategy of Adam in advance. Second, we study the effect of relaxing the full observation assumption on the complexity of computing winning strategies for Eve.

Regarding regret-minimizing strategies, we give algorithms to compute the strategies of Eve that ensure minimal regret against three classes of adversaries: (*i*) unrestricted, (*ii*) limited to positional strategies, or (*iii*) limited to word strategies. These results apply for quantitative games defined with the classical payoff functions Inf, Sup, LimInf, LimSup, mean payoff, and discounted sum.

For *partial-observation games*, we continue the study of energy and mean-payoff games started in 2010 by Degorre et al. We complement their decidability result for a particular problem related to energy games (the FIXED INITIAL CREDIT PROBLEM) by giving tight complexity bounds for it. Also, we show that *mean-payoff games* are undecidable for all versions of the mean-payoff function. Motivated by the latter negative result, we define and study several decidable sub-classes of mean-payoff games. Finally we extend the newly introduced *window mean-payoff objectives* to the partial observation setting. We show that they are conservative approximations of partial-observation mean-payoff games and we classify them according to whether they are decidable. Furthermore, we give a symbolic algorithm to solve them.

Acknowledgements

This work would not have been possible without the support of the *Fonds de la Recherche Scientifique* FNRS.

I owe many thanks to my thesis adviser, Jean-François Raskin, for accepting to fund the first two years of my PhD with his ERC grant and, more importantly, for teaching me to do and enjoy research. Special thanks also go to Paul Hunter for acting as my unofficial co-adviser. They both showed an amazing amount of patience with me and took the time to explain even the most simple parts of their every argument so that I could follow.

I am thankful to my family for their support and visits. In particular, I want to thank my Mom and Dad for coming to drink (too much) Belgian beer with me in Brussels. Also, thanks to my brothers, Carlos and Eduardo, for visiting during my first year in Europe. Last, but not least, thanks to *mijn lekker ding*, Kim, for being my local family.

During the last four years I have met several other PhD students and post-docs who influenced me in some way. I have been very lucky to be able to befriend those of you who have, at some point, spent some time at ULB: Alexander, Mathieu, Mahsa, Romain (a.k.a. J.-C.), Manu, Gilles, Lorenzo, Axel, Benjamin, Ocan, Stéphane, Noemie, Aaron, Thierry, Van-Anh, Mick, Luc, Ismaël, Nathan, Nico, etc. Very special and warm thanks go to the francophone subset of the above list for helping me with my french and with the cultural shock that goes with living abroad.

Finally, I would like to thank people who took the time to read earlier versions of this document and helped me find (most?) typos and mistakes: Honza, Gilles, and Jean-François.

Contents

Chapter 1

Introduction

In this chapter we will introduce the reader to a part of the field of verification. First, we will describe what kinds of systems we are aiming to formally verify, and, eventually, synthesis in an automated fashion. We will then focus particularly on the latter: the automatic generation of correct-by-construction reactive systems. Our goal, in that second section, is to highlight the connections between the synthesis task and that of finding a winning strategy for a player in a game. Finally, we will list the contributions to the game-theoretic foundations of reactive synthesis developed in this dissertation.

1.1 Reactive Systems

Reactive (computer) systems are systems that maintain a continuous interaction with their environment. They are ubiquitous in modern society. For example, embedded controllers used in cars and planes, system software, and device drivers, are all reactive systems. Although the correct functioning of reactive systems is often safety or economically critical, their design and implementation is difficult and error-prone. To support the design of such systems, in a way that ensures correctness, mathematical logic and automata-theoretic methods have been studied for decades. Seminal works in the 1950s by Kleene [Kle56] and in the 1960s by Büchi, Rabin, among others, on automata theory and logic have started this successful research track [Rab69] that focuses on verification of reactive systems. Nowadays, companies such as Intel and Facebook are realizing the importance of being able to mathematically prove the correctness of their systems, leading to an increase in the profile of the field of *formal verification.*

Academically, research on formal verification ranges from contributions to the theoretical foundations of the area to the development of efficient algorithms. In practice, Intel, Microsoft, Airbus, and more recently Facebook and Amazon, all have dedicated verification teams and industry-level verification tools.[1]

To further argue the need for formal verification, we present some examples of how it is useful in real-world scenarios.

Real-world examples and applications. In 2015 a group of researchers attempted to formally verify the correctness of several sorting algorithms im-

[1]See, for instance, the infer tool developed by Facebook: `http://fbinfer.com/`

plemented in Java. Instead of successfully verifying them, they found a bug in the default sorting algorithm inside the Android software development kit and the OpenJDK Java development kit [dGRdB+15] using formal verification tools. Sorting is, arguably, one of the most basic tasks that any modern computerized system should be able to do. The fact that this bug had escaped code reviewers and testing, and yet was diagnosed by a formal verification tool, highlights the need for the use of such tools to become a default in software (and hardware) development processes.

As an application, we focus on *artificial intelligence*. Artificial intelligence is slowly becoming the biggest buzzword when it comes to developments related to computer science. Autonomous vehicles and board game algorithms [SHM+16] are examples of where artificial intelligence is being applied. It has, however, become obvious that behavior learned by robots must be paired with *safety guarantees*. In [AOS+16], the authors argue that one of the most interesting research directions regarding artificial intelligence is to find ways to make sure agents do not damage themselves or their environments while learning. We posit that deciding if a learning algorithm has that property, is in fact a verification problem. Further, constructing learning algorithms which are safe-by-construction might be related to reactive synthesis (which we will define in the sequel).

Model checking. One of the most successful techniques developed by the formal verification community is *model checking*: given a model of the system at hand together with a correctness specification, determine whether all the possible behaviors of the model satisfy the specification. Advancements in model checking have been recognized as important contributions by the computer science community. This recognition has taken the form of two Turing awards: the first in 1996, awarded to Amir Pnueli for his work on temporal logic; the second in 2007, shared by Clarke, Emerson, and Sifakis for their seminal works on model checking itself.

One weakness of classical model checking is the fact that it is Boolean. That is, a system either satisfies a specification or it does not [AK86, BK08]. Recently, *quantitative verification* has started to receive more attention as a way to address questions such as: "What is the probability of battery power dropping below minimum?" [Kwi07]. Models analyzed by such quantitative methods are usually variants of finite automata or graphs with costs or rewards labelling their transitions (respectively, edges).

1.2 Synthesis and Games

A more ambitious goal than that of model checking (or proving that a given system is correct in general) is to synthesize reactive systems from their specification and ensure correctness by construction. The main theoretical tool which has been used to model this task is *game theory*. The required system, or *controller*, is seen as a player choosing actions corresponding to outputs expected from it, while its *environment* is modelled as a second player choosing actions that correspond to the uncontrollable inputs which are fed to the system. An infinite-duration game is then played by the two players on an arena which models the possible configurations of the whole system. The winner is then determined by whether the infinite sequence of configurations meets the desired

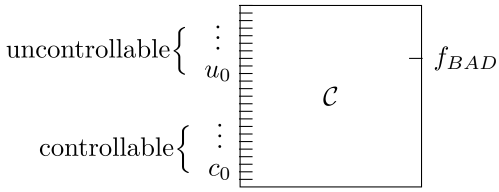

Figure 1.1: An example of a succinct safety specification for the reactive synthesis task.

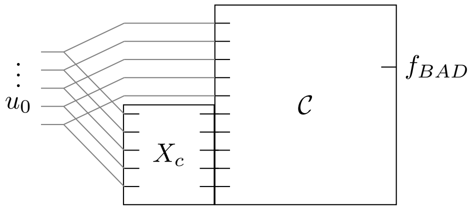

Figure 1.2: A solved instance of a safety-specification reactive synthesis task.

specification. If the controller has a set of rules—also known as a *strategy*—to ensure he wins against any behavior of the environment, this may be implementable as the required system. The reason the latter is not always possible is that a winning strategy may require unbounded amounts of memory to keep track of, for instance, all actions witnessed so far. This motivates our interest in the memory requirements of players in different games.

To give the reader a more concrete example of what we mean by synthesis of reactive systems, we briefly sketch the rules for the **Synthesis Competition** [JBB+14] organized since 2013 in the context of the SYNT workshop. Suppose we are given a *sequential circuit* $\mathcal{C}$ with a set of inputs and a single output, such as the one depicted in Figure 1.1. The output of $\mathcal{C}$ is *high*, or equal to 1, if and only if an error has occurred. In other words, the output of $\mathcal{C}$ signals that the sequence of inputs it has been fed resulted in it reaching some internal bad configuration. Further suppose that the inputs of the circuit are partitioned into controllable and uncontrollable (again, like in Figure 1.1). We would like to, automatically, obtain a sequential circuit X_C which generates the controllable inputs, as a function of the uncontrollable ones, so that the error output will never be high. If such a circuit exists, we can compose it with $\mathcal{C}$ (see Figure 1.2) and be sure that, regardless of the sequence of uncontrollable inputs fed into X_C and $\mathcal{C}$, the circuit never reaches a bad configuration.

Seminal works on synthesis, *e.g.* [Tho95], are mostly based on models which assume that all the information is available to the system (full observation), that the specification is Boolean (a trace of the system is either good or bad), and that the environment is completely antagonistic (a zero-sum situation). As is the case for model checking, if we wish to apply reactive synthesis to real-world scenarios, more robust models need to be considered. This dissertation focuses on such extensions: *quantitative games with partial observation* and *regret minimization in quantitative games* as a non-zero-sum solution concept.

We will now further elaborate on the connection between game theory and verification, and reactive synthesis in particular.

Game theory. Game theory [OR94] is a branch of mathematics which is applied in various application domains including economics, social sciences, biology, political sciences, as well as engineering. It attempts to model strategic situations where several individuals are interacting, and it tries to predict what will be the decisions taken by these individuals in a given situation. In this framework, a strategic situation is called a game, each individual taking part in this game is called a player, and their decisions are called strategies. For instance, an auction on eBay can be seen as a game where the bidders are the players, the prices at which the players will stop to increase their offers are their strategies. The systematic study of game theory started in 1944 with the book "Theory of Games and Economic Behavior" by John von Neumann and Oskar Morgenstern [vNM07]. This book mainly focused on strictly competitive situations in which only two individuals interact, also known as zero-sum games. Another important step forward in the development of game theory, around 1950, has been the introduction of the concept of Nash equilibrium (named after John Forbes Nash, who proposed it) together with a proof of their existence [N$^+$50]. Informally, a profile of strategies is a Nash equilibrium if no player can do better by unilaterally changing her strategy. This important notion allows us to study multi-player non-zero-sum games. It has been argued, however, that for some classical games, Nash-equilibria do not correspond to 'rational' behavior of the players [HP12]. This has motivated the study of alternative non-zero-sum solution concepts such as regret minimization.

Game theory for reactive synthesis. More recently, game theoretic concepts have been introduced in computational models [Tho95]. The basic framework that extends computational models with concepts from game theory is that of so-called two-player zero-sum games played on graphs. In such games there are two players, and plays are infinite paths constructed by the two players who alternate in taking moves (traversals of edges). The winning condition is given by a set of good infinite paths that one of the players tries to enforce against the other player, who tries to build a path outside that set. Many problems in verification and design of reactive systems can be modelled in this framework [AHK02]. One of the applications of games in computer-aided design is reactive synthesis. Given a model of the system to control (given as a graph) and a control objective (for instance, to prevent the system to reach some bad configurations), the controller synthesis problem asks to build a controller (a program) that interacts with the system and ensures that the control objective is enforced. In such a setting it is easy to see how modelling the controller as one player and its system (also often called the environment) as an adversary, captures the fact that the controller must react to the actions of the system in a way that the control objective is enforced. As expected, to solve this problem, zero-sum games played on graphs are adequate models [Tho95]. In such games, vertices model configurations of the system, moves of player 1 model actions of the controller, moves of player 2 model the uncontrollable actions of the system, and the original control objective induces a set of paths in the graph which are winning for player 1. A winning strategy for player 1 (the controller) is an abstract form of a control program that enforces the control objective no matter how the system chooses its actions.

1.3 Contributions

The present dissertation summarizes our contributions to the theory of reactive synthesis—via game theory—in two directions. First, we have started the study of regret minimization in quantitative games played on graphs. Second, we have continued the study of quantitative games with partial observation. We elaborate on these two directions below.

Regret. Quantitative games are played on finite directed graphs with rational weights on their edges. In the classical zero-sum setting, one of the players attempts to maximize the value of a *payoff function* that aggregates the sequence of weights seen along a play, while the other player strives to minimize the same value. For several payoff functions, these games have been studied and shown to be generalizations of well-known Boolean games. Most of the contributions for the game-theoretical foundations of reactive synthesis, in the context of quantitative games, are for zero-sum games. That is, the objective of the player that models the system is to maximize the value of the game while the objective of his opponent is to minimize this value (just as described above). This is a worst-case assumption: because the cooperation of the environment cannot be assumed, we postulate that it is *antagonistic*.

Equilibria [N$^+$50] are, arguably, the most well-studied non-zero-sum solution concept (see, *e.g.*, [Kri03, Ves06, Rou09, RP15]). More recently, Halpern and Pass have advocated for alternative solution concepts which include *regret minimization* [HP12]. We have started a study of the computational complexity of regret minimization in quantitative games. More specifically, we have focused on synthesizing strategies for the controller which minimize its regret when assuming the environment is of a specific class (namely, memoryless strategies or word strategies, which will defined formally later). We have also considered the most general case in which the environment is completely unrestricted. One can argue that regret is not the best solution concept for the task of reactive synthesis, indeed it does not guarantee, against all possibilities, a specific bound. However, as we will see in the sequel, it transpires that regret-free strategies are also worst-case optimal. Also, non-zero-sum solution concepts have recently been gaining attention, for example, as a tool to synthesis controllers with guarantees conditioned on assumptions on the environment [BCH$^+$16]. Furthermore, it can be shown that regret-free strategies are in fact also optimal in the worst-case sense [HPR16b, HPR16a]. Additionally, regret-free strategies have the advantage of being superior to any alternative strategy when compared against any witnessed behavior of the environment, that is, of course, the definition of regret.

Unexpectedly, regret minimization against word strategies of the environment has been shown to be a generalization of a restricted kind of determinization for Boolean automata on infinite words [HPR16b]. Our results, presented in this dissertation, thus can also be seen as an approach to tackle unsolved problems [CDH10] related to quantitative automata on infinite words and their determinization.

Partial Observability. A game is said to have (asymmetric) partial observation if Eve is not given the exact configuration of the game after every turn. That is to say, whenever the current state of the game changes, instead of informing Eve about the new state, a set of possible states—containing the actual state of the game—is given to her.

Games with partial observation arise naturally in economics and have a rich and well-developed theory. However, in computer science we only have preliminary results about games with partial observation [Rei84, CDHR06, DDG$^+$10]. A large part of the results on games apply only for models with full observation. Unfortunately, this hypothesis is often unrealistic: when some digital device has to control a physical system, it acquires information about the state of the system through sensors with finite precision. This results in the controller having partial observation about the state of the system to control. In this scenario, we want to construct so-called observation-based strategies, that is, strategies that only depend on the information acquired by the sensors. Partial observability also arises naturally in multi-component systems: the individual components have only a partial view on the state of the other components. Typically, in a shared memory system, processes have access to their local variables and the global variables but not the local variables of other processes.

Games played on graphs with partial observation are conceptually and computationally harder than games with full observation. For example, optimal strategies in partial-observation games with simple reachability objectives require the use of randomization [CDHR06] while pure memoryless strategies exist for reachability objectives under full observability. Games with partial observation and co-Büchi objectives are undecidable while, under full observability, games with the same objectives are solvable in polynomial time [GO10, CD12]. Also, deciding the winner for quantitative objectives (mean-payoff and energy games) is also undecidable under partial observation [DDG$^+$10]. A lot of work remains to be done in order to better understand these undecidability results. For the decidable cases, the usual technique is based on subset constructions that transform the game graph with partial observation into a game graph with full observation.

In the literature, there are several decidability vs. undecidability results about games with partial observation, *e.g.* [Rei84, DDG$^+$10]. Nevertheless, our understanding of the decidability frontier is far from being complete, and there is a lack of general results and conceptual tools. For example, for mean-payoff games with partial observation, the problem has been shown undecidable when objectives are defined using strict thresholds, but the case for non-strict thresholds was left open in [DDG$^+$10]. In this work, we look for general properties that are sufficient (in some games) to ensure that positive decidability results can be transferred from a full-observation game to the original partial-observation game. To be concrete, we focus on quantitative games. These games are important in the context of controller synthesis: we may want, for example, a strategy that ensures that energy consumed by the system is below some threshold per time unit, or that the time separating every request (of a client) from its grant (by the server) is minimal.

For classes of games that have been shown undecidable, it is often possible to identify sub-classes that are decidable. For example, it was shown in [DDG$^+$10] that mean-payoff games with visible weights form a decidable subclass. We show in the sequel that we can substantially extend this result by showing that all mean-payoff games for which the *belief* graph (its subset construction) only contains simple cycles whose concrete paths can all be deemed 'good' or 'bad', in a well-defined sense that we will formalize later, form a decidable subclass. For this and other sub-classes, we investigate the exact complexity of recognizing such instances and the complexity of synthesizing winning strategies for such games. To obtain, decidability results for quantitative games, we also study variants for the definition of the objectives. Particularly, instead of considering constraints over the mean payoff along an infinite play, we study objectives, recently proposed in [CDRR13] and called *window mean-payoff objectives*, that ask that it is always true along the play that the next k steps (a window of size k) have a mean payoff above a given threshold.

Chapter 2

Preliminaries

In this chapter, we will go through most of the mathematical notations which are used in later chapters. We will also derive some intermediate results that will be, sometimes implicitly, used in the sequel.

2.1 Notations and Conventions

We assume the reader is familiar with basic discrete mathematics and computational complexity theory. Whenever possible, we adhere to the following notational conventions. Use lowercase letters such as $a, b, \ldots$ for elementary objects; uppercase letters such as $A, B, \ldots$ for sets of elementary objects; uppercase letters in calligraphic font such as $\mathcal{A}, \mathcal{B}, \ldots$ for structures composed of other objects; lowercase Greek letters $\alpha, \beta, \ldots$ for functions and sequences; and, finally, uppercase Greek letters for sets of the latter.

The symbols $|$ and $:$ will be used interchangeably to separate elements of a set and the properties these must satisfy, variable quantification and formulas, etc. Both of them should be read as "such that".

Sets and functions. For a set S, we denote by $\mathcal{P}(S)$ the set of subsets of S (also referred to as the *power set* of S).

Let $\alpha : A \to B$ be a function mapping elements from the set A to elements from the set B. The set A is the *domain* of α and we denote it by $\mathsf{Domain}(\alpha)$. The set B is the *range* of α and we denote it by $\mathsf{Range}(\alpha)$. Let $\beta : A \nrightarrow B$ be a partial function mapping a strict subset of A to elements from B. We denote by $\mathsf{supp}(\beta)$ the *support* of β: the set $A' \subset A$ for which the function is defined. We may sometimes favor using functions, instead of partial functions, and just add a distinct new element to B which will be the value assigned by it to all $a \in A$ for which it would, otherwise, not be defined. For example, we replace β by $\kappa : A \to (B \cup \{\bot\})$ which is such that $\kappa(a) = \beta(a)$ if β is defined for a and $\kappa(a) = \bot$ otherwise. The support of κ is the set $\{a \in A \mid \kappa(a) \neq \bot\}$.

Sequences and tuples. We will write a sequence of elements as a comma-separated list in parenthesis. For instance, we could list all non-negative integers in order: $(0, 1, 2, \ldots)$. Alternatively, we might use angle brackets, *i.e.* $\langle \cdot \rangle$, instead of parentheses for improved readability, (for example, in cases where we have sequences of sequences). As a third possibility, we may omit the parentheses and commas altogether in order to avoid notation saturation.

The usual sets of numbers. We denote by $\mathbb{R}$ the set of *real numbers*; $\mathbb{Q}$ the set of *rational numbers*; $\mathbb{N}$ the set of *natural numbers*—including 0; and $\mathbb{N}_{>0}$ the set of *positive integers*. We will also use the Boolean numbers $\mathbb{B} := \{0, 1\}$.

The usual notation for intervals of real numbers will be sometimes used. That is, we denote by (a, b) the set $\{x \in \mathbb{R} \mid a < x < b\}$; by $[a, b)$, the set $\{x \in \mathbb{R} \mid a \leq x < b\}$; by $(a, b]$, the set $\{x \in \mathbb{R} \mid a < x \leq b\}$; and by $[a, b]$, the set $\{x \in \mathbb{R} \mid a \leq x \leq b\}$.

Quasi-orderings and Vectors of Naturals. For a (possibly infinite) set of numbers N, a relation $R \subseteq N \times N$ is said to be a *quasi-order* if it is reflexive (*i.e.*, for all $n \in N$ it holds that $(n, n) \in N$) and transitive (*i.e.*, for all $n_1, n_2, n_3 \in N$ if we have $(n_1, n_2) \in N$ and $(n_2, n_3) \in N$ then $(n_1, n_3) \in N$). Typically, one writes nRm instead of $(n, m) \in R$.

An example of a quasi-order for the set of natural numbers $\mathbb{N}$ is $\leq$. If R is a quasi-order for a set N and, additionally, for any infinite sequence $n_1, n_2, \ldots$ such that $n_1 \in N$ there are two positions $i, j \in \mathbb{N}_{>0}$ such that $i < j$ and $n_i R n_j$, then we say R is a *well-quasi-order*.

We denote by $\mathbb{N}^d$ the set of all natural-number vectors of dimension d. For two vectors $\mathbf{a} = (a_d, \ldots, a_1), \mathbf{b} = (b_d, \ldots, b_1) \in \mathbb{N}^d$ we write $\mathbf{a} \leq \mathbf{b}$ if and only if $a_i \leq b_i$ for all $1 \leq i \leq d$. That is, $\leq$ is the product order or component-wise order. Dickson's Lemma tells us that $\leq$ is a well-quasi-order for $\mathbb{N}^d$.

Directed graphs. A directed graph is a pair (V, E) consisting of a set V of *vertices* and a set $E \subseteq V \times V$ of *edges* with a direction associated to them. That is, an edge $(u, v) \in E$ is considered to 'leave' u and 'arrive' at v. In this dissertation, all considered graphs are directed. Thus, henceforth we often omit the adjective 'directed' and write digraph or just graph.

Let $\mathcal{G} = (V, E)$ be a directed graph and consider $(u, v) \in E$. The vertex v is said to be a *direct successor* of u and u is a *direct predecessor* of v. We write $\mathsf{deg}^+(u)$ for the *outdegree* of u, that is, the number of direct successors of u, and $\mathsf{deg}^-(u)$ for the the *indegree* of u, the number of direct predecessors of u. A vertex $u \in V$ has a *self-loop* if $(u, u) \in E$. A *path* is a sequence of vertices $v_0 v_1 \ldots$ where v_{i+1} is a direct successor of v_i, for all $i \geq 0$; it is said to be *simple* if $v_i \neq v_j$ holds for all $0 \leq i < j$. If a path contains a vertex v, we sometimes say it 'visits' or 'reaches' v; if it contains the sub-sequence uv then we say it 'takes' or 'traverses' the edge (u, v). If there is a finite path from $u \in V$ to $v \in V$ in the graph, *i.e.* there is a path $u v_1 \ldots v_n v$, then we say v is a successor of u and u is a predecessor of v; alternatively, we may say v is *reachable* from u. If a vertex $v \in V$ has no direct successor, we call it a *sink*; if it has only one direct successor and a self-loop, we say it is *trapping*. A *cycle* is a finite path $\chi = v_0 \ldots v_n$ with $v_0 = v_n$; it is said to be simple if $v_i \neq v_j$ holds for all $0 \leq i < j < n$. We refer to a finite path $\lambda = v_0 \ldots v_{n-1} v_n \ldots v_m$, where $v_0 \ldots v_{n-1}$ is a finite path and $v_n \ldots v_m$ is a cycle, as a *lasso*.

Players and their preferences. In this work we will be interested in games played by two players only. We will refer to the first one (the one we are, in some sense, “supporting”) as Eve and to the second one as Adam. In most cases, Eve will be interested in maximizing some value that Adam will try to minimize.

2.2 Languages, Automata, and Topology

In this dissertation we study systems, and thus games, with infinite behaviors. We will repeatedly make use of infinite sequences studied in formal language theory and topology. Hence, we need to introduce some of their notation.

Languages. Consider a (possibly infinite) set A of *symbols*. A *word* on A is a sequence $a_0a_1\dots$ of elements from A. The special *empty word* is denoted by ε. We sometimes refer to a set of symbols, such as A, as an *alphabet* and to symbols as *letters*. Given a finite word $\alpha = a_0 \dots a_n$ and a (possibly infinite) word $\beta = a'_0a'_1 \dots$, we denote their *concatenation*: $a_0 \dots a_n a'_0 \dots$ by $\alpha \cdot \beta$. We denote by A^* the set of all finite words on A, that is to say the set

$$\{a_0 \dots a_n \mid a_i \in A \text{ for all } 0 \leq i \leq n\}$$

of finite sequences of elements from A. The set

$$\{a_0 \dots \mid \forall i \geq 0 : a_i \in A\}$$

of infinite words on A we denote by A^ω.

Given $B \subseteq A^*$ and C a set of (infinite or finite) words on A, we denote by $B \cdot C$ the concatenation of B and C. That is, the set consisting of all words constructed by concatenating a word from B to a word from C:

$$\{\beta \cdot \kappa \mid \beta \in B, \kappa \in C\}.$$

Regular and Omega-regular Languages. A set $L \subseteq A^*$ of finite words on A is a *language* over A. A special set of languages over A, the *regular* languages, is defined recursively as follows:

- The empty language $\varnothing$ and the empty word language $\{\varepsilon\}$ are both regular languages.
- Any singleton language $\{a\}$, for $a \in A$, is a regular language over A.
- For any regular language B over A, B^* is also a regular language over A.
- For any two regular languages B, C over A, their union and concatenation is also a regular language over A.
- No other language over A is regular.

A set $L \subseteq A^\omega$ of infinite words on A is referred to as an *ω-language* over A. An ω-language L over A is ω-regular if any of the following hold:

- $L = B^\omega$, where $B \subseteq A^*$ is a non-empty regular language not containing the empty word.
- $L = B \cdot C$, where $B \subseteq A^*$ is a regular language and $C \subseteq A^\omega$ is an ω-regular language.
- $L = B \cup C$, where $B, C \subseteq A^\omega$ are both ω-regular.

Omega-automata. Automata over infinite words are very similar to their finite-word counterparts: they consist of a finite set of states, transitions labelled with symbols, and an initial state. However, since the words accepted by them do not end, *i.e.* they are infinite, there are no 'final' states. Thus, more involved acceptance conditions are needed.

Formally, an *automaton* is a tuple (Q, q_0, A, Δ) where Q is a finite set of *states*, $q_0 \in Q$ is the *initial state*, A is a finite alphabet (of actions), and $\Delta \subseteq Q \times A \times Q$ is the *transition relation*. We assume that Δ is *total*, in the following sense: for all $(q, a) \in Q \times A$, there is $q' \in Q$ such that $(q, a, q') \in \Delta$. If Δ is functional, *i.e.* for all $q \in Q$ and all $a \in A$ there is a unique $q' \in Q$ such that $(q, a, q') \in \Delta$, then we say the automaton is *deterministic* and write $\delta(q, a)$ to denote q'. Given a set $S \subseteq Q$ and a letter $a \in A$ we denote by

$$\mathsf{post}_a(S) := \{q \in Q \mid \exists p \in S : (p, a, q) \in \Delta\}$$

the set of a-successors of S. In a slight abuse of notation, we write $\mathsf{post}_a(q)$ instead of $\mathsf{post}_a(\{q\})$ to improve readability.

Consider an automaton $\mathcal{A} = (Q, q_0, A, \Delta)$. A *run* of $\mathcal{A}$ over an infinite word $a_0 a_1 \dots \in A^\omega$ is a sequence $q_0 a_0 q_1 a_1 \dots$ such that $(q_i, a_i, q_{i+1}) \in \Delta$ for all $i \geq 0$. A run $\varrho = q_0 a_0 q_1 \dots$ is then said to be *accepting* if some property—regarding the states which appear infinitely often in ϱ—is fulfilled. The automaton $\mathcal{A}$ then accepts an infinite word α if it has an accepting run over α. We call the set of all infinite words which are accepted by $\mathcal{A}$ the language of $\mathcal{A}$ and denote it by $\mathcal{L}_\mathcal{A}$. We also say that $\mathcal{A}$ 'recognizes' $\mathcal{L}_\mathcal{A}$. Let $\varrho = q_0 a_0 \dots$ be a run of $\mathcal{A}$. The set of states which appear infinitely often in ϱ is defined as follows:

$$\mathsf{OccInf}(\varrho) := \{p \in Q \mid \forall i \geq 0, \exists j \geq i : q_j = p\}.$$

We write $\varrho[i]$ to denote the $(i+1)$-th state, that is q_i, in the sequence. Given indices $i, j \in \mathbb{N}$ such that $i \leq j$, we write $\varrho[i..j]$ for the *infix* $q_i a_i \dots a_{j-1} q_j$, $\varrho[..i]$ for the *prefix* $q_0 a_0 \dots a_{i-1} q_i$, and $\varrho[j..]$ for the *suffix* $q_j a_j \dots$ A run prefix $\pi = q_0 a_0 \dots a_{n-1} q_n$ ending in state $\mathsf{last}(\pi) = q_n$ is said to have length $n+1$, denoted $|\pi| = n + 1$.

Büchi, co-Büchi, parity, and Streett automata. Once more, let us consider an automaton $\mathcal{A} = (Q, q_0, A, \Delta)$. We will now describe several *acceptance conditions* and recall their expressive power. The following Boolean *payoff functions* can be used to define acceptance conditions for automata, *i.e.* to determine whether a run $\varrho = q_0 a_0 \dots$ is accepting. One can view a Boolean payoff function as the indicator function of a *payoff set* of infinite runs. For convenience, instead of defining a function $\mathbf{Val} : (Q \cdot A)^\omega \to \mathbb{B}$ we will define $\mathbf{Val}^{-1}(1)$.

- The Reachability function is defined for a set $T \subseteq Q$ of *target states* as:

$$\mathsf{Reach}^{-1}(1) := \{\text{run } q_0 a_0 \dots \mid \exists i \geq 0 : q_i \in T\}.$$

- The Safety function is defined for a set $U \subseteq Q$ of *unsafe states* as follows:

$$\mathsf{Safe}^{-1}(1) := \{\text{run } q_0 a_0 \dots \mid \forall i \geq 0 : q_i \notin U\}.$$

- The Büchi function is defined for a set of *accepting* or *Büchi* states $B \subseteq Q$ as:

$$\mathsf{Buchi}^{-1}(1) := \{\text{run } \varrho \mid \mathsf{OccInf}(\varrho) \cap B \neq \varnothing\}.$$

- The co-Büchi function is defined for a set of *rejecting* or *co-Büchi* states $B \subseteq Q$:
$$\mathsf{coBuchi}^{-1}(1) := \{\text{run } \varrho \mid \mathsf{OccInf}(\varrho) \cap B = \varnothing\}.$$
- The parity function is defined for a *priority function* $p : Q \to \mathbb{N}$ as:
$$\mathsf{parity}^{-1}(1) := \left\{\text{run } \varrho \mid \min_{q \in \mathsf{OccInf}(\varrho)} p(q) \text{ is even}\right\}.$$
- Finally, the Streett function is defined for a finite set of *Streett pairs* $\{(E_i, F_i) \mid i \in I\}$ where $E_i, F_i \subseteq Q$ for all $i \in I$. Its payoff set $\mathsf{Streett}^{-1}(1)$ is equal to
$$\{\text{run } \varrho \mid \forall i \in I : E_i \cap \mathsf{OccInf}(\varrho) \neq \varnothing \text{ or } F_i \cap \mathsf{OccInf}(\varrho) = \varnothing\}.$$

The are other classical payoff functions such as *Rabin* and *Muller* but we do not make use of them here.

We refer to an automaton with a payoff function **Val**, from the list above, as an **Val** automaton. For instance, the parity automaton $\mathcal{B} = (Q, q_0, A, \Delta, p)$ accepts a word $\alpha = a_0 a_1 \ldots$ if and only if it has a run $\varrho = q_0 a_0 \ldots$ on α such that $\mathsf{parity}(\varrho) = 1$. Incidentally, we refer to the number $\max(\mathsf{Range}(p))$ as the *index* of the parity function p.

Intuitively, a Büchi automaton with alphabet A has as its language the set of infinite words which have some *liveness* property, that is, some event represented by a Büchi state occurs infinitely often. Co-Büchi automata have a dual acceptance condition. This can be thought of as bad events occurring only finitely often (in at least one run of the automaton). Parity and Streett automata capture, in different ways, the idea that (except for a finite number of times) bad events must always be trumped by a good event. In the case of parity, any bad event—namely the occurrence of an odd parity—can either: take place a finite number of times, or a smaller and even priority must also occur an infinite number of times (to trump it). For Streett automata, the acceptance condition can be thought of as imposing a conjunction of "conditioned obligations": for all $i \in I$, if an element from F_i is seen infinitely often, then some element from E_i must also be seen infinitely often.

Parity and Streett automata are known to be more expressive than Büchi and co-Büchi automata when restricted to being deterministic. In fact, the following is well-known about ω-regular languages and automata on infinite words.

Proposition 2.1 (From [Tho95, PP04])**.** *For any alphabet A and ω-language L over A, the following are equivalent:*

- *L is ω-regular.*
- *There exists a Büchi automaton $\mathcal{A}$ such that $\mathcal{L}_{\mathcal{A}} = L$.*
- *There exists a deterministic parity automaton $\mathcal{A}$ such that $\mathcal{L}_{\mathcal{A}} = L$.*

■

Of particular interest to us in the context of the present dissertation is the fact that any non-deterministic Büchi automaton can be *determinized* into a parity automaton. In other words, for a given Büchi automaton, one can construct a deterministic parity automaton with exactly the same language. More formally:

Proposition 2.2 (Determinization of omega-automata [Saf88, Saf92, Pit07])**.** *Given a Büchi automaton $\mathcal{A} = (Q, q_0, A, \Delta, B)$, there is an algorithm which yields a deterministic parity automaton $\mathcal{A}' = (Q', q_0', A, \Delta', p)$ such that $|Q'|$ is of size $2^{\mathcal{O}(|Q| \log |Q|)}$ and with parity index polynomial with respect to $|Q|$.* ■

Borel hierarchy. Let A be a (possibly infinite) alphabet. The *Borel hierarchy* of subsets of A^ω is inductively defined as follows.

- $\Sigma_1^0 = \{W \cdot A^\omega \mid W \subseteq A^*\}$ is the set of *open subsets* of A^ω.
- For all $n \geq 1$, $\Pi_n^0 = \{A^\omega \setminus L \mid L \in \Sigma_n^0\}$ consists of the complement of sets in Σ_n^0.
- For all $n \geq 1$, $\Sigma_{n+1}^0 = \{\bigcup_{i \in \mathbb{N}} L_i \mid \forall i \in \mathbb{N} : L_i \in \Pi_n^0\}$ is the set obtained by countable unions of sets in Π_n^0.

A map $f : A \to B$ is said to be *Borel measurable* if $f^{-1}(X)$ is Borel for any open subset X of B.

2.3 Quantitative Payoff Functions

As the title of this work implies, we will focus mainly on non-Boolean payoff functions. We will now define some classical functions of the form

$$\mathbb{Q}^\omega \to (\mathbb{R} \cup \{-\infty, +\infty\}).$$

Formally, for an infinite sequence of rationals $\chi = x_0 x_1 \ldots$ we define:

- the Inf (Sup) payoff, is the minimum (maximum) rational seen along the sequence:
$$\mathsf{Inf}(\chi) := \inf\{x_i \mid i \geq 0\}$$
and
$$\mathsf{Sup}(\chi) := \sup\{x_i \mid i \geq 0\};$$
- the LimInf (LimSup) payoff, is the minimum (maximum) rational seen infinitely often:
$$\mathsf{LimInf}(\chi) := \liminf_{i \to \infty} x_i$$
and, respectively, we have that
$$\mathsf{LimSup}(\chi) := \limsup_{i \to \infty} x_i;$$
- the *mean-payoff* value of a sequence, *i.e.* the limiting average rational, defined using lim inf or lim sup since the running averages might not converge:
$$\underline{\mathsf{MP}}(\chi) := \liminf_{k \to \infty} \frac{1}{k+1} \sum_{i=0}^{k} x_i$$
and
$$\overline{\mathsf{MP}}(\chi) := \limsup_{k \to \infty} \frac{1}{k+1} \sum_{i=0}^{k} x_i.$$

In words, $\underline{\mathsf{MP}}$ corresponds to the *limit inferior* of the average of increasingly longer prefixes of χ while $\overline{\mathsf{MP}}$ is defined as the *limit superior* of χ.

Another payoff function we consider is the *discounted sum*. Given a sequence of rationals $\chi = x_0 x_1 \ldots$ of length $n \in \mathbb{N} \cup \{\infty\}$, the *discounted sum* is defined for a rational discount factor $\lambda \in (0, 1)$ as follows:

$$\mathsf{DS}_\lambda(\chi) := \sum_{i=0}^{n} \lambda^i x_i.$$

We remark the above payoff functions together with the Boolean functions—Büchi, parity, etc.—are all Borel measurable. Note that, since they map sequences of rationals to real numbers (or infinity) then it suffices to show that, for all $a \in \mathbb{R}$, the set of sequences with value above a is Borel. Formally, we have that:

Proposition 2.3. *Let $\triangleright \in \{>, \geq\}$. For all $a \in \mathbb{R}$, for any Boolean or quantitative payoff function* **Val**, *the set $\{\chi = x_0 x_1 \ldots \mid \mathbf{Val}(\chi) \triangleright a\}$ is Borel.* ■

Indeed, it is easy to use the definition of the payoff functions we have presented thus far, and convince oneself that the above holds. For example, for $a \in \mathbb{R}$ and $\triangleright$ set to $\geq$, the payoff function $\underline{\mathsf{MP}}$ yields:

$$\left\{ x_0 x_1 \cdots \in \mathbb{Q}^\omega \mid \bigcap_{i \in \mathbb{N}_{>0}} \bigcup_{j \in \mathbb{N}} \bigcap_{k \geq j} \frac{1}{k+1} \sum_{\ell=0}^{k} x_\ell \geq a - \frac{1}{j} \right\}$$

which is clearly Borel. It follows that all Boolean and quantitative payoff functions defined in this chapter are Borel-measurable functions.

Proposition 2.4. *All Boolean and quantitative payoff functions are Borel measurable.* ■

Prefix independence. A payoff function **Val** is said to be *prefix independent* if for any two sequences of rationals $\chi = x_0 x_1 \ldots, \chi' = x'_0 x'_1 \cdots \in \mathbb{Q}^\omega$ the following holds:

$$\left(\exists i \geq 0, \forall j \geq i : x_j = x'_j\right) \implies \mathbf{Val}(\chi) = \mathbf{Val}(\chi').$$

In other words, any two sequences with the same infinite suffix (starting from any point onwards) will have the same value. This can be seen as the payoff function "not caring" about the prefix of the sequence. Hence the name, prefix independent.

2.4 Computational Complexity

Throughout this dissertation, we follow notation and definitions from [GJ79] and [Pap03] for concepts regarding computational complexity. We regard algorithms which have polynomial worst-case running time as 'efficient'. Thus, we shall provide polynomial-time reductions when proving hardness results. Furthermore, we use *big-O* notation—*i.e.* $\mathcal{O}(\cdot)$—to describe the limiting behavior of functions (as was done in Proposition 2.2).

As we work with games played on weighted structures, let us comment on the format of the input of their decision problems. Unless explicitly stated otherwise, all weights labelling considered structures are given in binary. Furthermore, parameters for payoff functions—such as the discount factor λ required for the discounted sum function—are also given as input and in binary. Thus, an algorithm with worst-case running time $\mathcal{O}(\lambda)$ is of *pseudo-polynomial* running time. That is, polynomial in the numeric value of λ yet exponential in the size of its representation.

2.5 Recurrent Problems

We are interested in finding algorithms to determine the winner of games, as well as their computational complexity. In the following chapters, we shall present several reductions from known problems—complete or hard for some complexity class—to other problems. Some of the problems we reduce from, appear in more than one chapter. We present the most common ones here. Their relevance and similarity to the games we study will become clear once we formally define them.

2.5.1 Quantified Boolean formulas

A *fully quantified Boolean formula* is a formula in quantified propositional logic where every variable is quantified, either existentially or universally. For example, consider the following formula in which x, y, and z are all either existentially or universally quantified:

$$\exists x \forall y \exists z : (x \wedge y) \vee \neg z.$$

Such a formula is, therefore, either true or false. The QBF Problem consists in determining whether a given formula is true. It is known that this problem is Pspace-complete [SM73, GJ79].

The QBF problem is often rephrased as a game between Eve and Adam. In this game, the two players take turns (following the order of the quantifiers in the formula) to choose values for each variable from the given formula. In particular, Eve chooses the truth value for existentially quantified variables while Adam does so for universally quantified ones. The winner of the game is determined by the truth value of the formula, after all variables have been assigned a value. Eve wins the game if the formula is true while Adam wins if it is false.

One can, without loss of generality, assume that an instance of the QBF problem is always given in a very particular form. Formally, the input for the problem is a Boolean formula $\Psi = \exists x_0 \forall x_1 \dots \mathcal{Q} x_n(\Phi)$, where $\mathcal{Q} \in \{\exists, \forall\}$ and Φ is a Boolean formula expressed in *conjunctive normal form* (CNF). The game then begins by Eve choosing a value for x_0. Then, Adam responds by choosing a value for x_1, and so on and so forth.

2.5.2 Counter machines

A *Minsky machine* consists of a finite set of control states Q, initial and final states $q_I, q_F \in Q$, a set of counters C, and a finite set of instructions which act on the counters. Namely, inc_k increases the value of counter k by 1, dec_k decreases the same value by 1. Additionally, $0?_k$ serves as a *zero check* on counter k which blocks if the value of counter k is not equal to 0. More formally, the transition relation δ contains tuples (q, ι, k, q') where $q, q' \in Q$ are source and target states respectively, ι is an instruction from $\{inc, dec, 0?\}$ which is applied to counter $k \in C$. We focus here on *deterministic Minsky machines*, *i.e.* for every state $q \in Q$ either

- δ has exactly one outgoing transition, which is an increase instruction, *i.e.* $(q, \iota, \cdot, \cdot)$ with $\iota = inc$; or
- δ has exactly two transitions: a decrease $(q, dec, k, \cdot)$ and a zero check $(q, 0?, k, \cdot)$ instruction.

We denote by $|\mathcal{M}|$ the size of $\mathcal{M}$. Formally, $|\mathcal{M}| := |Q| + |\delta|$.

A *configuration of* $\mathcal{M}$ is a pair (q, v) of a state $q \in Q$ and a valuation $v : C \to \mathbb{N}$. (Note that we consider the variant of Minsky machines which guards all decreases with zero checks. Hence, all counters will have only non-negative values at all times.) A *run* of $\mathcal{M}$ is a finite sequence $\varrho = (q_0, v_0)\delta_0 \ldots \delta_{n-1}(q_n, v_n)$ such that $q_0 = q_I$, $v_0(k) = 0$ for all $k \in C$, and v_{i+1} is the correct valuation of the counters after applying δ_i to v_i for all $1 \leq i \leq n$. A run $(q_0, v_0)\delta_0 \ldots \delta_{n-1}(q_n, v_n)$ is *halting* if $q_n = q_F$ and it is *m-bounded* if $v_i(k) \leq m$ for all $0 \leq i \leq n$ and all $k \in C$.

We present three problems for Minsky machines with n counters, or nCMs, which are of interest to us. The first one will be used to obtain undecidability results.

Problem (The halting problem)**.** Given a 2CM $\mathcal{M}$, decide whether $\mathcal{M}$ has a halting run. ■

The halting problem was shown to be undecidable for Minsky machines with 2 or more counters by Minsky himself [Min67]. To do so, he constructs, from a Turing machine, a 2CM which halts if and only if the original machine halts.

Problem (f-Bounded halting problem)**.** Given a 2CM $\mathcal{M}$, decide whether $\mathcal{M}$ has an $f(|\mathcal{M}|)$-bounded halting run. ■

This bounded version of the halting problem is decidable. However, if f is a fast-growing function, then it is complete for the corresponding level of the fast-growing complexity classes [SS12, Sch16].

Finally, we consider a *promise problem* for 4CMs with an additional final *rejecting state* q_R. A run of such a machine is considered halting if, as before, it reaches a final state q_F or q_R. A halting run is then *rejecting* if it reaches q_R and *accepting* otherwise.

Problem (Deciding problem)**.** Given a 4CM guaranteed to have a halting run, decide if the halting run is accepting or rejecting. ■

The standard reduction from Turing machines to 4CMs, via finite state machines with two stacks, is readily seen to be constructible in polynomial time. If we start with a Turing *machine that always halts*, then in polynomial time we obtain a 4CM that always halts. As determining whether the former accepts or rejects is complete for the set of all decidable problems (*i.e.* R-complete), we conclude the Deciding problem for 4CMs is R-complete under polynomial reductions (decidability follows from the fact that we can simulate it and wait to see which state is reached).

Chapter 3

Quantitative Games

The main mathematical object studied throughout this dissertation is that of *two-player quantitative games* played on finite structures. In its most general form, these games are played on a finite directed graph with rational numbers labelling the edges: a directed weighted graph. We shall call the structure on which a quantitative game is played a *(weighted) arena*.

3.1 Games Played on Graphs

In this section, we shall first present all the definitions required in order to state well-known results on quantitative games played on graphs. We generally follow the definitions from [AG11]. Furthermore, we will recall some of these classical results since they will be used in several parts of this dissertation.

Quantitative games are played by two players: Eve and Adam. We will partition the vertices of an arena into those owned by Eve and those owned by Adam.

Definition (Weighted arena)**.** A *(weighted) arena* (or WA, for short) is a tuple $\mathcal{G} = (V, V_\exists, v_I, E, w)$ where (V, E) is a finite digraph, $V_\exists \subseteq V$ is the set of vertices belonging to Eve, $v_I \in V$ is the initial vertex, and $w : E \to \mathbb{Q}$ is a rational weight function assigning weights to the edges of the graph. ■

Since we will focus on infinite paths in arenas, we will assume that the underlying digraph of any arena has no sinks. We depict vertices owned by Eve (*i.e.* those in $V_\exists$) with squares and vertices owned by Adam (*i.e.* those in $V \setminus V_\exists$) with circles. We denote the maximum absolute value of a weight in an arena by $w_{\max}$. (See Figure 3.1 for an example of a weighted arena.)

A game played on a weighted arena by Eve and Adam proceeds in rounds as follows. Initially, the 'current vertex' is v_I, that is the initial vertex. From the current vertex u, if $u \in V_\exists$ then Eve chooses a direct successor v of u, otherwise Adam chooses a direct successor v of u. The process is then repeated from v. This interaction determines an infinite path in the arena. We shall call such an infinite path, a *play*. More formal definitions for these concepts follow.

Definition (Plays and prefixes)**.** A *play* in an arena $(V, V_\exists, v_I, E, w)$ is an infinite path in the digraph (V, E). In other words, a play is an infinite sequence of

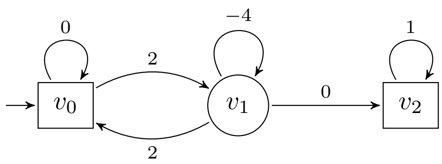

Figure 3.1: Example weighted arena with an Eve vertex, v_0, on the left and an Adam vertex, v_1, on the right. In this arena, $w_{\max}$ is equal to 4.

vertices $\pi = v_0 v_1 \ldots$ where $v_0 = v_I$ and $(v_i, v_{i+1}) \in E$ for all $i \geq 0$; a *play prefix* is then a finite path starting from v_I. ■

In the weighted arena depicted in Figure 3.1, examples of plays are v_0^ω, $v_0 \cdot v_1^\omega$, and $(v_0 \cdot v_1)^\omega$. An example of a sequence of vertices which is not a valid play is $v_0 v_1 v_2 \cdot v_0^\omega$ since there is no edge from v_2 to v_0 and therefore the sequence is not a path.

Definition (Strategies)**.** Consider an arena $\mathcal{G} = (V, V_\exists, v_I, E, w)$. A *strategy for* Eve (respectively, Adam) in $\mathcal{G}$ is a function that maps play prefixes ending with a vertex v in $V_\exists$ $(V \setminus V_\exists)$ to a direct successor of v. ■

Consider once more the arena from Figure 3.1. A sample strategy $\sigma : V^* \cdot V_\exists \to V$ for Eve is: from v_2 always play to v_2, and for any play prefix $\varrho \cdot v_0$ play v_1 if the length of ϱ is even and v_0 otherwise. An example of a strategy $\tau : V^* \cdot (V \setminus V_\exists)$ for Adam could be play from v_1 to v_1 always.

We say a strategy for Eve σ has *memory* m if there are: a non-empty set M with $|M| = m$, an element $m_0 \in M$, and functions $\alpha_u : M \times V \to M$ and $\alpha_o : M \times V_\exists \to V$ such that for any play prefix $\varrho = v_0 \ldots v_n$, we have $\sigma(\varrho) = \alpha_o(m_n, v_n)$, where m_n is defined inductively by $m_{i+1} = \alpha_u(m_i, v_i)$ for $i \geq 0$. The memory of a strategy of Adam is defined analogously. A strategy is said to have *finite memory* if $m \in \mathbb{N}_{>0}$. The tuple $(M, m_0, \alpha_u, \alpha_o)$ is sometimes referred to as the *Mealy machine realizing* σ (see, *e.g.*, [Gel14] and references therein).

Definition (Sets of strategies)**.** Consider an arena $\mathcal{G} = (V, V_\exists, v_I, E, w)$. We denote by $\mathfrak{S}_\exists(\mathcal{G})$ and $\mathfrak{S}_\forall(\mathcal{G})$ the sets of all strategies for Eve and, respectively, for Adam in $\mathcal{G}$; by $\mathfrak{S}_\exists^m(\mathcal{G})$ and $\mathfrak{S}_\forall^m(\mathcal{G})$ the sets of all strategies with memory m for both players. ■

Also, if $\mathcal{G}$ is clear from the context, we shall omit it.

A strategy for either player with memory 1 is said to be *positional* or *memoryless*. Let $(M, m_0, \alpha_u, \alpha_o)$ realize a strategy for one of the players. Observe that when $|M| = 1$ we then have that $M = \{m_0\}$ and thus α_o does not depend on its first parameter (since it is always m_0). A positional strategy can then be expressed as a function from vertices to vertices as follows:

$$\sigma : V_\exists \to V \text{ (or } \tau : (V \setminus V_\exists) \to V, \text{ respectively).}$$

A play $\pi = v_0 \ldots$ in an arena $\mathcal{G}$ is said to be *consistent* with a strategy σ (respectively, τ) for Eve (Adam) if for all $i \geq 0$ it holds that $v_i \in V_\exists$ ($v_i \notin V_\exists$) implies that:

$$\sigma(\langle v_j \rangle_0^i) = v_{i+1} \; (\tau(\langle v_j \rangle_0^i) = v_{i+1}).$$

Definition (Outcome of strategies)**.** Consider an arena $(V, V_\exists, v_I, E, w)$. Given strategies σ and τ for Eve and Adam, respectively, we denote by $\pi^v_{\sigma\tau}$ the unique play starting from $v \in V$ that is consistent with σ and τ. ■

In the $\pi^v_{\sigma\tau}$ notation, if v is omitted we assume it is v_I.

While fixing a strategy for each player results in a unique play induced by them, fixing a strategy for only one of them yields a set of plays consistent with it. One way to represent the latter set is to "combine" the weighted arena with a given strategy. More formally,

Definition (Product of an arena and a strategy)**.** Consider an arena $\mathcal{G} = (V, V_\exists, v_I, E, w)$ and a strategy σ for Eve in $\mathcal{G}$ realized by the Mealy machine $(M, m_0, \alpha_u, \alpha_o)$. We denote by $\mathcal{G} \times \sigma$ their *product*, that is the arena $(V \times M, \varnothing, (v_I, m_0), E', w')$ where:

- E' has an edge $((t, m), (v, n))$ if only if $(t, v) \in E$, $\alpha_u(m, t) = m$, and $t \in V_\exists \implies \alpha_o(m, t) = v$;
- w' maps $((t, m), (v, n))$ to $w(t, v)$ for all $(t, m), (v, n) \in V \times M$.

The product of $\mathcal{G}$ with a strategy for Adam is defined similarly. ■

Intuitively, the product of an arena and a strategy is a weighted arena in which Eve no longer controls any vertex since her choices have been fixed according to the strategy. If the strategy used finite memory, the resulting weighted arena is finite. Note that there is indeed a one-to-one correspondence between plays consistent with a strategy σ for a player in $\mathcal{G}$ and plays in $\mathcal{G} \times \sigma$.

So far, we have defined the type of arena on which the games we study will be played. Additionally, we have formalized the notion of strategy for each of the two players who will take part in the games. Now, we must assign a value to a play, so as to determine how much each player gains from having witnessed it. We shall do so by using *payoff functions* of the form $\mathbb{Q}^\omega \to \mathbb{R}$ (see Section 2.2 and Section 2.3).

Definition (Value of a play)**.** Consider an arena $\mathcal{G} = (V, V_\exists, v_I, E, w)$ and a payoff function $\mathbf{Val} : \mathbb{Q}^\omega \to \mathbb{R}$. The *value of a play* $\pi = v_0 v_1 \ldots$ in $\mathcal{G}$ is denoted by $\mathbf{Val}(\pi)$ and defined as:

$$\mathbf{Val}(w(v_0, v_1) w(v_1, v_2) \ldots).$$

■

For simplicity, we denote the value of a play starting from v and consistent with strategies σ and τ, for Eve and Adam respectively, as

$$\mathbf{Val}^v(\sigma, \tau) := \mathbf{Val}(\pi^v_{\sigma\tau}).$$

For the case when $v = v_I$ we write simply $\mathbf{Val}(\sigma, \tau)$, *i.e.* we omit v.

The above definition of the value of a play concludes the set of basic definitions required to determine a quantitative game. Indeed, what we refer to as a 'quantitative game' is merely a weighted arena together with a payoff function. Examples of such payoff functions are Büchi, parity, Inf, Sup, LimInf, and LimSup. (Note that for quantitative payoff functions such as LimSup, it is not the case that for any sequence of rationals χ we have $\mathbf{Val}(\chi) \in \mathbb{R}$. Indeed, their values might be $-\infty$ or $+\infty$ for some sequences of rationals. However, when applied to a sequence of rational weights obtained from a weighted arena, they are easily seen to be bounded by a function of $w_{\max}$.) For simplicity, we refer to the quantitative game consisting of arena $\mathcal{G}$ and payoff function $\mathbf{Val}$ as a $\mathbf{Val}$ game, *e.g.* an '$\underline{\mathsf{MP}}$ game' a.k.a. a mean-payoff game. Whenever the payoff function is understood from the context, we directly speak of an arena as being a game.

A *Boolean game* is a particular case of quantitative game. The payoff functions for such games are of the form $V^\omega \to \mathbb{B}$ and are obtained by adapting the Boolean payoff functions for automata runs, such as Büchi and parity (see Section 2.2), to sequences of vertices.

3.1.1 Winning condition

By far the most widely studied *solution concept* for quantitative games used for reactive synthesis is that of *winning strategies* (we shall define winning strategies briefly). In game theory, a solution concept is a rule predicting how a game will or should be played. As we are interested in how Eve should behave, we are looking for a solution concept for Eve. Intuitively, a winning strategy for her is a strategy which ensures some property regardless of what strategy Adam plays. This is an extremely robust concept—hence its popularity. In order to properly define it, we need some more notation.

Let $\mathcal{G}$ be a quantitative game with weighted arena $(V, V_\exists, v_I, E, w)$ and payoff function $\mathbf{Val}$. In order for a strategy for Eve in $\mathcal{G}$ to be declared 'winning', we need a *winning condition* (a.k.a. *objective*) for the game: a subset of the plays which are desirable (or winning) for Eve. In quantitative games, a winning condition is determined by a *threshold*. For instance, if we fix $\nu \in \mathbb{R}$ we can then say that a play π in $\mathcal{G}$ is winning for Eve if it has a value of at least ν, that is, if it holds that $\mathbf{Val}(\pi) \geq \nu$. Hence, we have a partition of the set of plays into winning and losing for Eve: the set $\{\pi \mid \mathbf{Val}(\pi) \geq \nu\}$ is winning for her, and the complement is losing for her. Two different, yet interrelated questions, arise from this definition. The first, asks, for a given winning condition, whether a player can ensure to win against any behavior of his adversary. The second, asks what is the maximal or minimal threshold for which he can do the latter. We focus here on the first question and deal with the second question in the following section.

Problem (Deciding a game)**.** Given a quantitative game with weighted arena $(V, V_\exists, v_I, E, w)$, payoff function $\mathbf{Val}$, and threshold ν, determine whether there exists a strategy σ for Eve in $\mathcal{G}$ such that, against any strategy τ for Adam in $\mathcal{G}$ it holds that $\mathbf{Val}(\sigma, \tau) \geq \nu$. If the latter holds, then the strategy of Eve witnessing it, is called a *winning strategy* for her. If Eve has a winning strategy in a game, she wins that game. ■

Similarly, a winning strategy for Adam is a strategy such that, against any strategy of Eve, the resulting play is not winning for Eve. If Adam has a winning strategy, we say he wins the game.

Note that, even if we can determine whether Eve wins a game, it is not immediate how to obtain a winning strategy for her. Further, determining the winner of a game might be 'easier', in terms of computational complexity, than to obtain a winning strategy. This motivates the following search problem.

Problem (Winning strategy synthesis)**.** Given a quantitative game with arena $(V, V_\exists, v_I, E, w)$, payoff function $\mathbf{Val}$, and threshold ν, if Eve wins the game output a strategy σ for her such that:

$$\inf_{\tau \in \mathfrak{S}_\forall} \mathbf{Val}(\sigma, \tau) \geq \nu.$$

■

Both problems stated above have been centered on Eve. A useful property of all quantitative games considered in this dissertation is that: if Eve does not have a winning strategy, then necessarily Adam has one. A game with this property is said to be *determined*. Let us formalize these claims.

Definition (Determinacy)**.** A quantitative game $\mathcal{G}$ is said to be determined if: either Eve has a winning strategy, or Adam has a winning strategy. ■

A quantitative game being determined can be seen as a kind of "quantifier swap" property which holds for the logic formulas stating the existence of winning strategies for the players. (We remark that, in general, it does not hold that $\neg(\exists A, \forall B : \varphi)$ implies $\exists B, \forall A : \neg\varphi$ and that is why determined games are so special.) In a determined quantitative game, deciding the game is equivalent to determining the winner of the game.

A very general result due to Martin [Mar75] implies that any quantitative game with a Borel winning condition is guaranteed to be determined. All winning conditions studied in the present dissertation can easily be shown to be Borel subsets of the set of all plays (see Proposition 2.3).

Proposition 3.1 (Borel determinacy [Mar75])**.** *For all weighted arenas $\mathcal{G} = (V, V_\exists, v_I, E, w)$, for all payoff functions* $\mathbf{Val}$*, for all thresholds $\nu \in \mathbb{R}$, if the corresponding set of winning plays $W \subseteq V^\omega$ is a Borel subset of the set of all plays in $\mathcal{G}$ then the* $\mathbf{Val}$ *game played on $\mathcal{G}$ is determined.* ■

The above result will prove to be extremely useful throughout this dissertation. More specific statements hold for different quantitative games, *e.g.* in mean-payoff games one of the two players has a winning positional strategy for every threshold. We recall these and other results regarding classical games in subsection 3.1.3. In the next section we will be interested in the maximal threshold for which Eve is guaranteed to have a winning strategy.

3.1.2 Values of a game

Previously, we have asked whether a player can enforce outcomes with value of at least a given threshold (or, respectively, at most a threshold). However, instead of fixing a threshold, we could directly define the *value of a game*. Since

Eve is attempting to maximize the value of the witnessed play and Adam is trying to do the opposite, we have the following two alternative values of a game in which the players are *antagonistic*:

$$\overline{\mathbf{aVal}^v(\mathcal{G})} := \sup_{\sigma \in \mathfrak{S}_\exists(\mathcal{G})} \inf_{\tau \in \mathfrak{S}_\forall(\mathcal{G})} \mathbf{Val}^v(\sigma, \tau), \text{ and}$$
$$\underline{\mathbf{aVal}^v(\mathcal{G})} := \inf_{\tau \in \mathfrak{S}_\forall(\mathcal{G})} \sup_{\sigma \in \mathfrak{S}_\exists(\mathcal{G})} \mathbf{Val}^v(\sigma, \tau).$$

It should be clear that the following relation among the two values trivially holds for all $v \in V$:

$$\underline{\mathbf{aVal}^v(\mathcal{G})} \leq \overline{\mathbf{aVal}^v(\mathcal{G})}.$$

Furthermore, one can show—using Borel determinacy or directly applying a result from [Mar98]—that if **Val** is bounded and Borel-measurable then both are equivalent.

Proposition 3.2 ((Unique) antagonistic value)**.** *For any quantitative game with arena $\mathcal{G} = (V, V_\exists, v_I, E, w)$ and bounded Borel-measurable payoff function* **Val***, for all $v \in V$ it holds that $\underline{\mathbf{aVal}^v(\mathcal{G})} = \overline{\mathbf{aVal}^v(\mathcal{G})}$.* ■

Henceforth, we shall refer to

$$\mathbf{aVal}^v(\mathcal{G}) := \underline{\mathbf{aVal}^v(\mathcal{G})} = \overline{\mathbf{aVal}^v(\mathcal{G})}$$

as the *antagonistic value* of a game $\mathcal{G}$ (and we will omit $\mathcal{G}$, as usual, if it is clear from the context, or v if it is assumed to be v_I).

Let us remark that all payoff functions considered in this work are in fact bounded and Borel-measurable. (Measurability was already argued in Proposition 2.4. All payoff functions used in this dissertation can be shown to be bounded by a function of $w_{\max}$ due to the finiteness of the arenas.) We thus consider the problem of computing the unique antagonistic value of a given game.

Problem (Computing the value of a game)**.** Given a quantitative game $\mathcal{G}$, output its antagonistic value $\mathbf{aVal}^{v_I}(\mathcal{G})$. ■

Following the definitions of the antagonistic value of a game and of the problem of deciding a game, one might be tempted to say that the former (an optimization problem) is harder than the latter. However, even if the antagonistic value of a game turns out to be μ, this does not imply that there is a strategy for Eve which actually achieves at least μ against any strategy for Adam. We will later see that for some classical quantitative games, this intuition does turn out to be correct and *worst-case optimal strategies* do exist for both players.

Definition (Worst-case optimal strategies)**.** In a quantitative game $\mathcal{G}$ with weighted arena $(V, V_\exists, v_I, E, w)$ and payoff function **Val**,

- a strategy σ for Eve is said to be *worst-case optimal (maximizing)* from $v \in V$ if it holds that $\inf_{\tau \in \mathfrak{S}_\forall} \mathbf{Val}^v(\sigma, \tau) = \mathbf{aVal}^v(G)$, and
- a strategy τ for Adam is said to be *worst-case optimal (minimizing)* from $v \in V$ if it holds that $\sup_{\sigma \in \mathfrak{S}_\exists} \mathbf{Val}^v(\sigma, \tau) = \mathbf{aVal}^v(G)$.

■

Thus far we have considered the case where Adam attempts to witness a play with a small value. One can also study the case in which both Eve and Adam *co-operatively* maximize the same value. We then have the *co-operative value* of a quantitative game $\mathcal{G}$ played on arena $(V, V_\exists, v_I, E, w)$ with payoff function **Val**:

$$\mathbf{cVal}^v(\mathcal{G}) := \sup_{\sigma \in \mathfrak{S}_\exists} \sup_{\tau \in \mathfrak{S}_\forall} \mathbf{Val}^v(\sigma, \tau)$$

where $v \in V$.

Definition (Co-operative optimal strategies)**.** In a quantitative game $\mathcal{G}$ with weighted arena $(V, V_\exists, v_I, E, w)$ and payoff function **Val**, a pair of strategies σ and τ for Eve and Adam, respectively, is said to be *co-operative optimal* from $v \in V$ if $\mathbf{Val}^v(\sigma, \tau) = \mathbf{cVal}^v(\mathcal{G})$. ■

We observe that this scenario can be reduced to a one-player game: a game in which Eve owns all the vertices.

We conclude this section on the values of a quantitative game by observing that, for Boolean games, computing the antagonistic values is not really an interesting problem. Indeed, by definition, we have that for any Boolean game $\mathcal{G}$ it holds that $\mathbf{cVal}(\mathcal{G}), \mathbf{aVal}(\mathcal{G}) \in \mathbb{B}$. Also note, for threshold $\nu = 0$ Eve always has a winning strategy and for any threshold $\nu > 1$ Adam always has a winning strategy. Furthermore, Eve wins the game for any threshold $0 < \nu < 1$ if and only if she wins for threshold $\nu = 1$. Thus, the only interesting question in these games is whether Eve wins the game with threshold 1. Henceforth, we shall implicitly assume, for Boolean games, a threshold $\nu = 1$.

3.1.3 Classical games

In this section, we will recall the definitions and properties of several classical quantitative games. First, we will define all the Boolean games used in the following chapters. We then focus on non-Boolean quantitative games.

Boolean games

Reachability, Safety, Büchi, co-Büchi, parity, and Streett games are defined using the corresponding Boolean payoff functions. An interesting property about most of the aforementioned games is that they are positionally determined. More formally:

Proposition 3.3 (Positional determinacy [AG11])**.** *Reachability, safety, Büchi, co-Büchi, and parity games are* positionally determined*: either Eve has a positional winning strategy or Adam has a positional winning strategy.* ■

Additionally, determining the winner of all but parity and Streett games, can be done in polynomial time.

Proposition 3.4 (Complexity of determining the winner [EJ99, AG11])**.** *Determining the winner of a reachability, safety, Büchi, or co-Büchi game can be done in polynomial time. Determining the winner of a parity game is in* NP $\cap$ coNP*. Determining the winner of a Streett game is* coNP*-complete.* ■

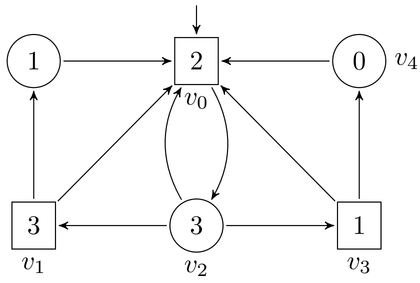

Figure 3.2: Parity game won by Eve. Vertices are labelled with their priorities.

There is no known polynomial-time algorithm to determine the winner of a parity game. Membership of the problem in NP∩coNP follows from the games being positionally determined, and the fact that one-player parity games are in polynomial time (hence, one can guess a strategy for a player and verify if the opponent can beat it). Its exact complexity is considered to be one of the most important open problems in formal verification. (It is known that the model checking problem for the modal μ-calculus reduces to determining the winner of a parity game.)

Example 1 (A parity-game example)**.** Consider the parity game depicted in Figure 3.2. From Proposition 3.3 it follows that if Eve has a strategy to ensure the minimal priority seen infinitely often is even, then she also has a positional strategy which ensures the same property. Note that the only vertices at which players have a choice of successor are v_1, v_2, and v_3. Let us consider the strategy for Eve which corresponds to the mapping $v_1 \mapsto v_0$ and $v_3 \mapsto v_4$. Again, from Proposition 3.3, we know that if Adam can beat this strategy for Eve (*i.e.* force a play consistent with it for which the minimal priority seen infinitely often is odd) then he has a positional strategy to do so. Clearly, if Adam plays to v_1 from v_2 then the outcome of the strategies will be the play $(v_0v_2v_1)^\omega$ with value 1 since the priorities of the vertices are, respectively, 2, 3, and 3. If instead Adam plays to v_3 from v_1, then the resulting outcome $(v_0v_2v_3v_4)^\omega$ has value 1—since the priority of v_4 is 0 and it is the minimal priority in the arena. Finally, if Adam plays to v_0 from v_3 then the outcome is (v_0v_2) and its value is 1 since their priorities are 2 and 3. We conclude that the described strategy for Eve is winning. ■

Remark (Vertex-based vs. edge-based objectives)*.* Note that we have defined Boolean games played on directed graphs by specifying sets of distinct vertices (*e.g.* unsafe vertices). However, we can also define them with distinct edges. It is easy to see that one can reduce one version to the other, and vice versa, in polynomial time. ■

Quantitative games

Games defined using the quantitative payoff functions from Section 2.3 are the main focus of this work. It is not hard to see that Inf games generalize safety games; as Sup does reachability; LimInf does co-Büchi; and LimSup does Büchi.

Conversely, for any threshold, one can reduce the quantitative game to the Boolean game it generalizes. For instance, for LimSup and a threshold ν we can mark as Büchi edges all of those with weight at least ν and then play a Büchi game on the resulting arena. Clearly, Eve wins in the new game if and only if she wins in the original one. It follows that, for any threshold, these four quantitative games are positionally determined and solvable in polynomial time. For Inf, Sup, LimInf, and LimSup, the above already implies that in those games there always are worst-case optimal strategies for both players. (Also co-operative optimal pairs of strategies.) For mean payoff and discounted sum, the same property was shown to be true in [EM79, ZP96]. Thus, we have:

Proposition 3.5 (Existence of optimal strategies)**.** *For all quantitative games with payoff function* Inf*,* Sup*,* LimInf*,* LimSup*, mean payoff, or discounted sum, the following hold:*

- *there exists* $\sigma \in \mathfrak{S}^1_\exists$ *which is worst-case optimal maximizing from all* $v \in V$*,*
- *there exists* $\tau \in \mathfrak{S}^1_\forall$ *which is worst-case optimal minimizing from all* $v \in V$*,*
- *there are* $\sigma \in \mathfrak{S}^1_\exists$ *and* $\tau \in \mathfrak{S}^1_\forall$ *which are co-operative optimal from all* $v \in V$*.*

■

In terms of the complexity of solving the games, for all but mean payoff and discounted sum, we have already argued polynomial-time algorithms exist. Based on the above result we then get the following:

Proposition 3.6 (Complexity of determining the winner)**.** *Determining the winner of a* Inf*,* Sup*,* LimInf*, or* LimSup *game can be done in polynomial time. Determining the winner of a mean-payoff or discounted-sum game is in* NP ∩ coNP*.* ■

A reduction from mean-payoff games to discounted-sum games due to Zwick and Paterson [ZP96] tells us that reducing the complexity upper bound of the winner determination problem for discounted-sum games would result in a new upper bound for the complexity of the same problem for mean-payoff games. Additionally, Jurdziński has established a reduction from parity to mean-payoff games [Jur98] which gives an analogue of the latter for these two games. It follows that improving the upper bound for that problem, for mean-payoff or discounted-sum games, would be a major development in the area of verification.

Although no polynomial-time algorithm is known to determine the winner of a mean-payoff or discounted-sum game, pseudo-polynomial algorithms have been discovered.

Proposition 3.7 (From [ZP96, BCD$^+$11])**.** *The antagonistic value of a mean-payoff game can be computed in pseudo-polynomial time, i.e. in polynomial time w.r.t.* $|V|$*,* $|E|$*, and* $w_{\max}$*. The antagonistic value of a discounted-sum game can be computed in pseudo-polynomial time, i.e. polynomial w.r.t.* $|V|$*,* $|E|$*,* $\log_2 w_{\max}$*, and* λ*.* ■

We have two final remarks on the antagonistic and co-operative values of all games considered above. It follows from Proposition 3.3 and Proposition 3.6 that they can be easily represented in binary (see [BCD$^+$11] and [HM15] for the details regarding mean payoff and discounted sum).

Proposition 3.8 (Representation of the values of a game)**.** *For all quantitative games, both its co-operative and antagonistic values, i.e.* **cVal** *and* **aVal***, are representable using a polynomial number of bits.* ■

Also, the co-operative value of a game is easy to compute.

Proposition 3.9 (From [AG11, CDH10, ZP96])**.** *The co-operative value of all quantitative games can be computed in polynomial time.* ■

We will now study a sample mean-payoff game.

Example 2 (A mean-payoff game)**.** Consider the weighted arena from Figure 3.1 and let us focus on the mean-payoff function. If Eve controlled all the vertices, then she would be able to force a play of the form $\varrho \cdot (v_0 v_1)^\omega$. Note that any such play has value 2—that is, regardless of the prefix ϱ since mean payoff is prefix independent. Since there is no other play with a higher value in this game, 2 is its co-operative value. We will now argue that the antagonistic value of the game is 0. Recall that in a mean-payoff game Eve always has memoryless worst-case optimal strategies. Since at v_2 she does not really have a choice, then the only options are for her to stay in v_0 forever or move to v_1 every time the play reaches v_0. If she moves to v_1 then Adam might stay in v_1 forever. The outcome of these two strategies has value -4. If she stays in v_0 then the play v_0^ω has value 0 and Adam cannot change that. Hence the antagonistic value is indeed 0 and a worst-case optimal strategy for Eve in this game is to stay in v_0 always. ■

Particulars of discounted-sum games. In the first part of this dissertation we study several quantitative games which include discounted-sum games. Most of the techniques we develop apply uniformly to all functions except for discounted sum. We will, therefore, make use of some particular properties of discounted sum games to design special algorithms for those games. We will shortly recall these properties.

We recall the definition of a stronger version of co-operative optimality in which we ask the strategy to take co-operative optimal choices from all vertices reached via a play prefix consistent with the strategy [Fae09]:

Definition (Strongly co-operative optimal strategies for Eve)**.** Consider a quantitative game $\mathcal{G}$ with weighted arena $(V, V_\exists, v_I, E, w)$ and discounted-sum payoff function. A strategy σ for Eve in $\mathcal{G}$ is said to be *strongly co-operative optimal* (SCO) if for any play prefix $\varrho = v_0 \dots v_n$ consistent with σ, and such that $v_n \in V_\exists$:

$$\sigma(\varrho) = v' \implies v' \in \mathbf{cOpt}(v_n),$$

where $\mathbf{cOpt}(u) := \{v \in V \mid (u,v) \in E \text{ and } \mathbf{cVal}^u(\mathcal{G}) = w(u,v) + \lambda \mathbf{cVal}^v(\mathcal{G})\}$. ■

We will now define a new type of strategy for Eve: *co-operative worst-case optimal* strategies. A strategy is of this type if it attempts to maximize the co-operative value while achieving at least the antagonistic value. More formally,

Definition (Co-operative worst-case optimal strategies for Eve)**.** Consider a quantitative game $\mathcal{G}$ with weighted arena $(V, V_\exists, v_I, E, w)$ and discounted-sum payoff function. A strategy σ for Eve in $\mathcal{G}$ is said to be *co-operative worst-case optimal* (CWO) if for any play prefix $\varrho = v_0 \dots v_n$ consistent with σ, and such that $v_n \in V_\exists$,

$$\sigma(\varrho) = v' \implies v' \in \mathbf{wOpt}(v_n)$$

and

$$w(v_n, v') + \lambda \mathbf{cVal}^{v'}(\mathcal{G}) = \max\{w(v_n, v'') + \lambda \mathbf{cVal}^{v''}(\mathcal{G}) \mid v'' \in \mathbf{wOpt}(v_n)\},$$

where $\mathbf{wOpt}(u) := \{v \in V \mid (u, v) \in E \text{ and } \mathbf{aVal}^u(\mathcal{G}) = w(u, v) + \lambda \mathbf{aVal}^v(\mathcal{G})\}$.
■

A useful observation used by Zwick and Paterson in [ZP96], and which is implicitly used throughout this work, is the following.

Remark (Bellman equations)*.* Consider a quantitative game with weighted arena $(V, V_\exists, v_I, E, w)$ and discounted-sum payoff function. For all $u \in V$ it holds that $\mathbf{cVal}^u(G) = \max\{w(u, v) + \lambda \mathbf{cVal}^v(G) \mid (u, v) \in E\}$ and

$$\mathbf{aVal}^u(\mathcal{G}) = \begin{cases} \max\{w(u, v) + \lambda \mathbf{aVal}^v(\mathcal{G}) \mid (u, v) \in E\} & \text{if } u \in V_\exists \\ \min\{w(u, v) + \lambda \mathbf{aVal}^v(\mathcal{G}) \mid (u, v) \in E\} & \text{else.} \end{cases}$$

■

Using the above, one can easily show that a strategy σ for Eve is worst-case optimal if and only if for all play prefixes $\varrho = \dots v_n$ consistent with it, we have that: if $v_n \in V_\exists$ then $\sigma(\varrho) \in \mathbf{wOpt}(v_n)$. In other words, making local worst-case optimal choices is sufficient and necessary to ensure the antagonistic value of the game (*cf.* [BMR14]). Also, the same holds for the co-operative value. That is, making local co-operative optimal choices is sufficient and necessary to ensure the co-operative value. One can then show that both strongly co-operative optimal and co-operative worst-case optimal strategies for Eve always exist.

Proposition 3.10 (Existence of SCO and CWO strategies)**.** *In all discounted-sum games there exist strongly co-operative optimal strategies and co-operative worst-case optimal strategies for Eve.* ■

Proof. Let $\mathcal{G} = (V, V_\exists, v_I, E, w)$ be a weighted arena and $0 < \lambda < 1$ a rational discount factor.

We will first define a new arena $\mathcal{G}'$ such that any strategy for Eve in $\mathcal{G}'$ is a SCO strategy for Eve in $\mathcal{G}$ and any SCO strategy for Eve in $\mathcal{G}$ is a valid strategy for her in $\mathcal{G}'$. More precisely, $\mathcal{G}'$ is a sub-arena of $\mathcal{G}$ obtained by removing from $\mathcal{G}$ all edges (u, v) leaving vertices of Eve, *i.e.* $u \in V_\exists$, for which it holds that $v \notin \mathbf{cOpt}(u)$. It follows from the Bellman equations for discounted-sum games that for every vertex u, there is at least one of its direct successors v, such that the edge (u, v) present in $\mathcal{G}$ is also present in $\mathcal{G}'$. Hence, $\mathcal{G}'$ has no sinks and Eve has at least one strategy in $\mathcal{G}'$. Furthermore, it is easy to show that $\mathcal{G}'$ indeed represents the set of all SCO strategies for Eve in $\mathcal{G}$ in the sense described above (see our argument above regarding local optimal choices being sufficient and necessary for a strategy to be optimal). Thus, there exists an SCO strategy for Eve in any discounted-sum game.

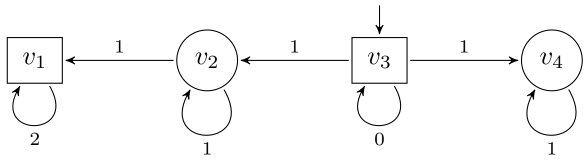

Figure 3.3: A sample weighted arena. With discounted-sum payoff function, the only CWO strategy for Eve is to play left.

For CWO strategies we proceed similarly. We define an arena $\mathcal{G}''$ such that any strategy for Eve in $\mathcal{G}''$ is a CWO strategy for her in $\mathcal{G}$ and any CWO strategy for her in $\mathcal{G}$ is a valid strategy for her in $\mathcal{G}''$. First, remove all choices of Eve which are not worst-case optimal locally. That is, remove all edges (u, v) such that $u \in V_\exists$ and $v \notin \mathbf{wOpt}(u)$. Next, from the remaining edges, remove all of her choices which are not co-operative optimal locally. It once more follows from the Bellman equations that the obtained arena $\mathcal{G}''$ has no sinks, and thus Eve has at least one strategy. Further, it is not hard to show any strategy for Eve in the remaining sub-arena is a CWO strategy for her in $\mathcal{G}$ and vice versa. $\square$

We will now study a simple discounted-sum game in order to better illustrate what SCO and CWO strategies are.

Example 3 (A discounted-sum game)**.** We focus on the discounted-sum game played on the arena shown in Figure 3.3. We do not instantiate a particular discount factor but just refer to it as λ. It is easy to see that the antagonistic value of the game is $\frac{1}{1-\lambda}$ while its co-operative value is $1 + \lambda + \lambda^2(\frac{2}{1-\lambda})$. Indeed, the play with the best discounted-sum value goes directly to v_1 and stays there.

Let us compare the types of strategies Eve has in this game. If Eve does not play from v_3 to v_2 from the first turn, then she is not able to achieve the co-operative value. Hence, her only SCO strategy is to play $v_3 \mapsto v_2$ and $\mathbf{cOpt}(v_3) = \{v_2\}$. (Note that she only really has a choice of successor at v_3 so a positional strategy for her is just a mapping from v_3 to a successor.) Since the antagonistic value of the game is $\frac{1}{1-\lambda}$, only strategies which always go to v_2 or v_4 from v_3 are worst-case optimal for her. In other words, $\mathbf{wOpt}(v_3) = \{v_2, v_4\}$. Finally, regarding CWO strategies, we note that although going from v_3 to v_4 does ensure the antagonistic value, it does not allow for a higher co-operative value like going to v_2 does (there, Adam could eventually move to v_1). Hence, the only CWO behaviour of Eve consists in moving to v_2. $\blacksquare$

3.2 Games Played on Automata

In this section we consider an alternative choice of arena in which the games we study can be played. Namely, we describe how Eve and Adam can take turns to build infinite runs in an automaton, and the relation of such games to games played on graphs.

A game played on an automaton (Q, q_0, A, Δ) with *action set* A proceeds in rounds. First, from the current state $p \in Q$ of the game, Eve chooses an action $a \in A$. Then, Adam selects a state q from $\mathsf{post}_a(p)$. The new state of the game then becomes q. This process is repeated *ad infinitum* and its outcome is an infinite run. Formally, a game consists of an automaton and a payoff function (as is the case for games played on directed graphs).

Quantitative games played on automata require we extend our definition of automata.

Definition (Weighted automata)**.** A *weighted automaton* is a tuple $\mathcal{A} = (Q, q_0, A, \Delta, w)$ where (Q, q_0, A, Δ) is a usual automaton and $w : \Delta \to \mathbb{Q}$ is a *weight function* that assigns rational weights to transitions. ■

Notions of determinism, runs, etc. are inherited from the usual automata for infinite words. As in graph games, let $w_{\max}$ denote the maximum absolute value of a transition weight in the automaton.

A weighted automaton $\mathcal{A} = (Q, q_0, A, \Delta, w)$ and a payoff function **Val** are, together, called a *quantitative automaton.* A quantitative automaton realizes a function from words to real numbers (this generalizes the notion of language for Boolean automata). More formally, for a word $\alpha = a_0 a_1 \ldots$, and a run $\varrho = q_0 a_0 \ldots$ of $\mathcal{A}$ on α, we denote by $\mathbf{Val}(\varrho)$ the value of the run, *i.e.*

$$\mathbf{Val}(w(q_0, a_0, q_1) w(q_1, a_1, q_2) \ldots).$$

We then write $\mathcal{A}(\alpha)$ for the supremum of the values of all its runs on the word.

In a quantitative game played on an automaton, a play is a run from the automaton and strategies are extended in the natural way from their graph game definitions. We remark that games played in graphs can be transformed into games played on automata. (This can be achieved, *e.g.*, by having states modelling vertices of Eve allow choice of successor depending on actions, and states corresponding to vertices of Adam transitioning to all direct successors with all actions.) The same is true in the opposite direction, from automata games to graph games. (Here one can, for instance, split transitions into two edges with the same weight as the original transition and adjust the weights to account for the doubling of path lengths.)

The usefulness of considering an alternative arena for quantitative games will become obvious in the sequel. Intuitively, since automata are assumed to have a total transition relation (see Section 2.2), Eve can play any action from any state. Our goal is to generalize our definition of a game so that Eve is only conscious of a set of possible states she could be at. Hence, if not all actions were available, this would give her some additional information. Formal definitions follow.

Definition (Partial-observation games)**.** A *(quantitative) game with partial observation* is a tuple $(Q, q_0, A, \Delta, w, \mathsf{Obs})$ together with a payoff function **Val** where (Q, q_0, A, Δ, w) is a weighted automaton and $\mathsf{Obs} \subseteq \mathcal{P}(Q)$ is a partition of Q into *observations.* ■

Pictorially, we depict states by circles and observations by dotted boxes around the states they contain (see Figure 8.1 and other examples in the second part of this dissertation). If $\mathsf{Obs} = \{Q\}$ we say the game is *blind*, if $\mathsf{Obs} = \{\{q\} \mid q \in Q\}$ we say it is a *full-observation* game.

Example 4 (A partial-observation game example)**.** Consider the reachability game with partial observation from Figure 3.4. When the game starts, Eve

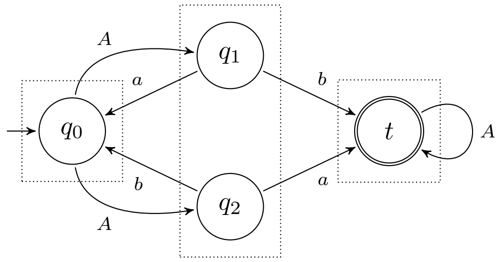

Figure 3.4: Partial-observation reachability game in which Eve does not have an observation-based winning strategy. The alphabet is $A = \{a, b\}$. The target states as depicted as double circles.

knows the current state of the game: q_0. She then plays some action from A and Adam selects a state from $\{q_1, q_2\}$ and reveals the observation in the middle to Eve. Since Eve does not know which state has been reached, she cannot base her choice of next action on that knowledge. Note that if she plays a from q_1 then the game is reset; the same is true if she plays b from q_2. Consider any strategy for Eve and let $\alpha = a_0 a_1 \ldots$ be the sequence of actions she plays when the middle observation is reached—if the middle observation is never again observed then she must have won the game. It is easy to see that, Adam, in order to keep the play from reaching t can choose to go to q_1 for all $i \geq 0$ such that $a_i = a$ and to q_2 otherwise. Hence, there is no winning observation-based strategy for Eve in the game. ■

Unless otherwise stated, in what follows we consider a fixed **Val** game $\mathcal{A} = (Q, q_0, A, \Delta, w, \mathsf{Obs})$. Note that in partial-observation games we can focus on observation-action sequences or on state-action sequences. In what follows we adapt the notions of plays, strategies, etc. from games played on graphs to partial-observation games played on automata.

Plays. A *(concrete) play* in $\mathcal{A}$ is an infinite sequence $\pi = q_0 a_0 q_1 \ldots$ such that $(q_i, a_i, q_{i+1}) \in \Delta$, for all $i \geq 0$. The set of plays in $\mathcal{A}$ is denoted by $\mathsf{Plays}(\mathcal{A})$ and the set of prefixes of plays ending in a state is written $\mathsf{Prefs}(\mathcal{A})$. The unique observation containing state q is denoted by $\mathsf{obs}(q)$. We extend $\mathsf{obs}(\cdot)$ to plays and prefixes in the natural way. For instance, we obtain the *observation sequence (or abstract play)* $\mathsf{obs}(\pi)$ of a play π as follows: $\mathsf{obs}(q_0) a_0 \mathsf{obs}(q_1) a_1 \ldots$; we then say π is a 'concretization' of $\mathsf{obs}(\pi)$. We also denote by $\mathsf{obs}^{-1}(\psi)$ the set of all concretizations of an abstract play (prefix) ψ. Finally, we denote by $\mathsf{obs}(\mathsf{Plays}(\mathcal{A}))$ and $\mathsf{obs}(\mathsf{Prefs}(\mathcal{A}))$ the set of all abstract plays and abstract prefixes, respectively.

Strategies. A *strategy for Eve* is a function $\sigma : \mathsf{Prefs}(\mathcal{A}) \to A$. A strategy σ for Eve is *observation-based* if for all prefixes $\pi, \pi' \in \mathsf{Prefs}(\mathcal{A})$, if $\mathsf{obs}(\pi) = \mathsf{obs}(\pi')$ then $\sigma(\pi) = \sigma(\pi')$. A prefix (or play) $\pi = q_0 a_0 \ldots$ is consistent with a strategy σ for Eve if $\sigma(\pi[..i]) = a_i$, for all $i \geq 0$. A *strategy for Adam* is a function $\tau : \mathsf{Prefs}(\mathcal{A}) \times A \to Q$. A prefix or play $\pi = q_0 a_0 \ldots$ is consistent with a strategy τ for Adam if $\tau(\pi[..i-1], a_{i-1}) = q_i$, for all $i > 0$. The memory used by a strategy, the Mealy machine realizing it, and the product of a game and an automaton are defined as for strategies in graph games.

Objectives. Winning conditions or objectives can be defined at two levels for games with partial observation. The first possibility is to have a set of good concrete plays for Eve $W \subseteq \mathsf{Plays}(\mathcal{A})$. The latter induces a set of good abstract plays $X \subseteq \mathsf{obs}(\mathsf{Plays}(\mathcal{A}))$ for her in the obvious way: the set of all observation sequences so that for no play $\pi \notin W$ we have $\mathsf{obs}(\pi) \in X$. Alternatively, we could directly define a set of good abstract plays $X \subseteq \mathsf{obs}(\mathsf{Plays}(\mathcal{A}))$ for Eve; the corresponding set of good concrete plays for her consists of all plays $\pi \in \mathsf{Plays}(\mathcal{A})$ such that $\mathsf{obs}(\pi) \in X$.

We say a strategy σ for Eve is *a winning strategy for her* in a game with objective $W \subseteq \mathsf{Plays}(\mathcal{A})$ if all plays consistent with σ are in W.

More definitions and properties of partial-observation games will be given in Chapter 8.

3.3 Non-Zero-Sum Games

We close this chapter with a brief comment on non-zero-sum games. In this chapter we have considered games in which, if a play is winning for Eve then it is not winning for Adam, and if it is winning for Adam then it is not winning for Eve. These are commonly referred to as *zero-sum games.* We will make copious use of results for zero-sum games. However, we will also consider non-zero-sum games. Namely, in Chapter 10 we make use of reachability games in which each player has their own target set of vertices they are trying to reach. Also, the first part of this dissertation is dedicated to regret minimization as a solution concept for non-zero-sum games. We defer the definitions and specifics of such games to the chapters where they are used.

Part I

Regret

Chapter 4

Background I: Non-Zero-Sum Solution Concepts

A fully antagonistic environment can sometimes be too coarse an abstraction of reality. In practice, the environment usually has its own goal which does not necessarily correspond to that of falsifying the specification of the reactive system. Nevertheless, this abstraction is popular because it is simple and sound. Against an antagonistic environment, the most commonly studied solution concept for Eve is that of a winning strategy. By definition, a winning strategy is in fact winning against any environment. That is, regardless of what the objective of the environment is and, further, regardless of whether he pursues it. A winning strategy may not exist, even if solutions for the modelled synthesis problem do exist when the objective of the environment is taken into account. Winning strategies, when they do exist, may also suffer from being sub-optimal because they are overcautious and do not exploit the fact that the environment attempts to satisfy an objective of its own in order to "do better".

In several recent works, new solution concepts have been studied for synthesis of reactive systems that take the objective of the environment into account or relax the fully adversarial assumption. In [BFRR14, CR15], the authors combined the classical formalism of two-player zero-sum games—where the environment is considered to be completely antagonistic—with *Markov decision processes*, a well-known model for decision-making inside a stochastic environment. They then propose a combined solution concept, which they call *beyond worst-case*. Essentially, they look for strategies which ensure some worst-case performance while maximizing the expected performance (against a given stochastic strategy for the environment, thus transforming the game into a Markov decision process). In [BRS15], the authors propose a novel notion of synthesis where the objective of the environment (and the assumption that he pursues it) can be captured using the game-theoretic concept of *admissible* strategies [BFK08, Ber07, BRS14]. In [HPR16b, HPR16a] we have focused on yet another classic game-theoretic solution concept recently popularized by Halpern and Pass: *regret minimization* [AG11, HP12].

Minimization of regret is a central concept in decision theory [Bel82]. It is also an important concept in game theory, see e.g. [ZJBP08] and references therein. *Iterated* regret minimization has been recently proposed by Halpern and Pass as a solution concept for *non-zero-sum* games [HP12]. There, it is applied to matrix games and not to game graphs. In a previous contribution, Filiot et al. have applied the iterated regret minimization concept to non-zero-sum games played on weighted graphs, for the shortest path problem [FGR10]. Restrictions on how Adam is allowed to play were not considered there. As we do not consider an explicit objective for Adam, we do not consider iteration of the regret minimization here.

In [HPR16b, HPR16a] we have studied strategies for Eve which *minimize her regret*. The regret of a strategy σ of Eve corresponds to the difference between the value Eve achieves by playing σ against Adam and the value she could have ensured if she had known the strategy of Adam in advance. Regret had not explicitly been used for games played on graphs before [FGR10]. The complexity of deciding whether a regret-minimizing strategy for Eve exists, and the memory requirements for such strategies change depending on two factors: (i) whether the payoff function is—or can be made—prefix independent, and (ii) what type of behavior Adam can use. Additionally, deciding whether Eve has a regret-free strategy turns out to be simpler than determining if she has a strategy which can ensure regret of at most r. Regarding the allowed behaviors for Adam, we have focused on three cases: arbitrary behaviors, positional behaviors, and time-dependent behaviors (otherwise known as *oblivious* environments). The latter class of regret games was shown in [HPR16b] to be related to the problem of determining whether an automaton has a certain form of determinism.

4.1 Regret Definition

Consider a fixed weighted arena $\mathcal{G} = (V, V_\exists, v_I, E, w)$, and payoff function $\mathbf{Val}$. Let $\Sigma_\exists \subseteq \mathfrak{S}_\exists$ and $\Sigma_\forall \subseteq \mathfrak{S}_\forall$ be sets of strategies for Eve and Adam respectively. Given $\sigma \in \Sigma_\exists$ we define the *regret of σ in $\mathcal{G}$ w.r.t. $\Sigma_\exists$ and $\Sigma_\forall$* as:

$$\mathbf{reg}^{\sigma}_{\Sigma_\exists,\Sigma_\forall}(\mathcal{G}) := \sup_{\tau \in \Sigma_\forall} (\sup_{\sigma' \in \Sigma_\exists} \mathbf{Val}(\sigma', \tau) - \mathbf{Val}(\sigma, \tau)).$$

We define the *regret of $\mathcal{G}$ w.r.t. $\Sigma_\exists$ and $\Sigma_\forall$* as:

$$\mathbf{Reg}_{\Sigma_\exists,\Sigma_\forall}(\mathcal{G}) := \inf_{\sigma \in \Sigma_\exists} \mathbf{reg}^{\sigma}_{\Sigma_\exists,\Sigma_\forall}(\mathcal{G}).$$

When $\Sigma_\exists$ or $\Sigma_\forall$ are omitted from $\mathbf{reg}(\cdot)$ and $\mathbf{Reg}(\cdot)$ they are assumed to be the set of all strategies for Eve and Adam.

In the current part of this dissertation we will study the computational complexity of the following decision problem.

Problem (The regret threshold, or r-regret, problem)**.** Given a quantitative game $\mathcal{G}$ with weighted arena $(V, V_\exists, v_I, E, w)$, sets of strategies $\Sigma_\exists \subseteq \mathfrak{S}_\exists$ and $\Sigma_\forall \subseteq \mathfrak{S}_\forall$, payoff function $\mathbf{Val}$, and regret threshold r, determine whether the regret of $\mathcal{G}$ is at most r. That is, the problem consists in deciding if

$$\mathbf{Reg}_{\Sigma_\exists,\Sigma_\forall}(\mathcal{G}) \leq r.$$

■

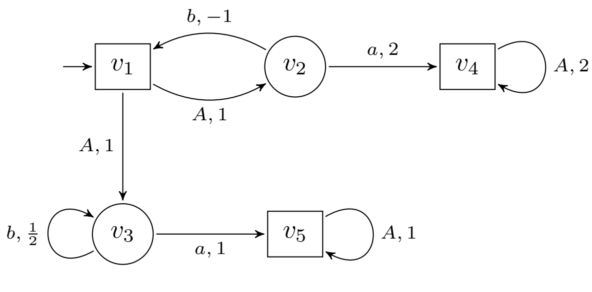

Figure 4.1: Mean-payoff game used to demonstrate how different sets of strategies for Adam yield different regret values.

4.2 Examples

Let us illustrate the notion of regret minimization on the example of Figure 4.1. (We do not use the letters labelling edges for the moment.) The game is played for infinitely many rounds and the value of a play for Eve is the long-run average of the values of edges traversed during the play (*i.e.* the mean-payoff value). In this game, Eve is only able to secure a mean payoff of $\frac{1}{2}$ when Adam is fully antagonistic. Indeed, if Eve (from v_1) plays to v_2 then Adam can force a mean-payoff value of 0, and if she plays to v_3 then the mean-payoff value is at least $\frac{1}{2}$. Note also that if Adam is not fully antagonistic, then the mean payoff could be as high as 2. Now, assume that Eve does not try to force the highest value in the worst-case but instead tries to minimize her regret. If she plays $v_1 \mapsto v_2$ then the regret is equal to 1. This is because Adam can play the following strategy: if Eve plays to v_2 (from v_1) then he plays $v_2 \mapsto v_1$ (giving a mean payoff of 0), and if Eve plays to v_3 then he plays to v_5 (giving a mean payoff of 1). If she plays $v_1 \mapsto v_3$ then her regret is $\frac{3}{2}$ since Adam can play the symmetric strategy. It should thus be clear that the strategy of Eve which always chooses $v_1 \mapsto v_2$ is indeed minimizing her regret.

In the following chapters, we will study three variants of regret minimization, each corresponding to a different set of strategies we allow Adam to choose from. The first variant is when Adam can play any possible strategy (as in the example above), the second variant is when Adam is restricted to playing memoryless strategies, and the third variant is when Adam is restricted to playing word strategies. To illustrate the last two variants, let us consider again the example of Figure 4.1.

A positional adversary. Assume now that Adam is playing memoryless strategies only. Then in this case, we claim that there is a strategy of Eve that ensures regret 0. The strategy is as follows: first play to v_2, if Adam chooses to go back to v_1, then Eve should henceforth play $v_1 \mapsto v_3$. We claim that this strategy has regret 0. Indeed, when v_2 is visited, either Adam chooses $v_2 \mapsto v_4$, and then Eve secures a mean payoff of 2 (which is the maximal possible value), or Adam chooses $v_2 \mapsto v_1$ and then we know that $v_1 \mapsto v_2$ is not a good option for Eve as cycling between v_1 and v_2 yields a payoff of only 0. In this case, the mean payoff is either 1, if Adam plays $v_3 \mapsto v_5$, or $\frac{1}{2}$, if he plays

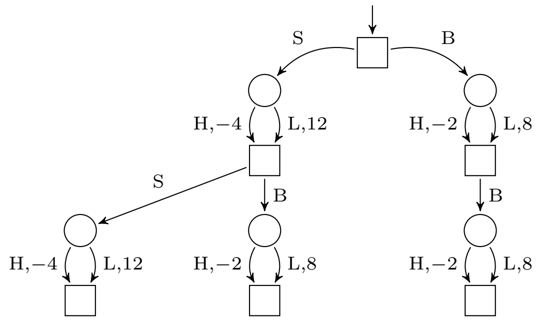

Figure 4.2: A game that models different investment strategies.

	HH	HL	LH	LL	Worst-case	Regret
SS	−7.7616	7.6048	7.9784	23.2848	−7.7616	3.8808
SB	−5.8408	3.7632	9.8392	19.4432	−5.8408	**3.8416**
BB	−3.8808	5.7232	5.9192	15.5232	**−3.8808**	7.7616

Table 4.1: The possible rate configurations for the rate of interests are given as the first four columns, then follows the worst-case performance and the regret associated to each strategy of Eve that are given in rows. Entries in bold are the values that are maximizing the worst case (strategy BB) and minimizing the regret (strategy SB).

$v_3 \mapsto v_3$. In all the cases, the regret is 0.

An eloquent adversary. Let us now turn to the restriction to word strategies for Adam. When considering this restriction, we use the letters that label the edges of the graph. A *word strategy* for Adam is a function $\tau : \mathbb{N} \to \{a, b\}$. In this setting Adam plays a sequence of letters and this sequence is independent of the current state of the game. It is more convenient to view the latter as a game played on a weighted automata in which Adam plays letters, hence the name *eloquent adversary*, and Eve responds by resolving non-determinism.

When Adam plays word strategies, the strategy that minimizes regret for Eve is to always play $v_1 \mapsto v_2$. Indeed, for any word in which the letter a appears, the mean payoff is equal to 2, and the regret is 0, and for any word in which the letter a does not appear, the mean payoff is 0 while it would have been equal to $\frac{1}{2}$ when playing $v_1 \mapsto v_3$. So the regret of this strategy is $\frac{1}{2}$ and it is the minimal regret that Eve can secure.

Note that the three different strategies give three different values in our example. This is in contrast with the worst-case analysis of the same problem (memoryless strategies suffice for both players).

Investment advice. We will now give a second example of the usefulness of regret minimization in quantitative games.

Consider the discounted-sum game depicted in Figure 4.2. It models the rentability of different investment plans with a time horizon of two periods. In the first period, it can be decided to invest in treasure bonds (B) or to invest in the stock market (S). In the former case, treasure bonds (B) are chosen for two periods. In the latter case, after one period, there is again a choice for either treasure bonds (B) or stock market (S). The returns of the different investments depend on the fluctuation of the rate of interests. When the rate of interests is low (L) then the return for the stock market investments is equal to 12 and for the treasure bonds it is equal to 8. When the interest rate is high (H) then the returns for the stock market investments is equal to -4 and for the treasure bonds it is equal to -2. To model time and take into account the inflation rate, say equal to 2 percent, we consider a discount factor $\lambda = 0.98$ for the returns. In this example, we make the hypothesis that the fluctuation of the rate of interests is *not* a function of the behavior of the investor. This means that this fluctuation rate is either one of the following four possibilities: HH, HL, LH, LL. This corresponds to Adam playing a *word strategy* in our terminology. The discounted sum of returns obtained under the 12 different scenarios are given in Table 4.1.

Now, assume that you are a broker and you need to advise one of your customers regarding his next investment. There are several ways to advise your customer. First, if your customer is strongly *risk averse*, then you should be able to convince him that he has to go for the treasure bonds (B). Indeed, this is the choice that *maximizes the worst case*: if the interest rates stay high for two periods (HH) then the loss will be -3.8808 while it will be higher for any other choice. Second, and maybe more interestingly, if your customer tolerates some risk, then you may want to keep him happy so that he will continue to ask for your advice in the future! Then you should propose the following strategy: first invest in the stock market (S) then in treasure bonds (B) as this strategy *minimizes regret*. Indeed, at the end of the two investment periods, the actual interest rates will be known and so your customer will evaluate your advices *ex-post*. So, after the two periods, the value of the choices made *ex-ante* can be compared to the best strategy that could have been chosen knowing the evolution of the interest rates. The regret of SB is at most equal to 3.8416 in all cases and it is minimal: the regret of BB can be as high as 7.7616 if LL is observed, and the regret of SS can be as high as 3.8808.

Finally, let us remark that if the investments are done in financial markets that are subject to different interest rates, then instead of considering regret minimization against word strategies, then we could consider the regret against all strategies.

4.3 Prefix Independization

It is well-known that LimInf, LimSup, $\underline{\mathsf{MP}}$, and $\overline{\mathsf{MP}}$ are prefix independent. Often, the arguments that we develop in the following chapters work uniformly for these four measures because of their prefix-independent property. Although Inf and Sup are not prefix independent, in the sequel we apply a simple transformation to the game and encode Inf into a LimInf objective, and Sup into a LimSup objective. The transformation consists of encoding in the vertices of the arena the minimal (maximal) weight that has been witnessed by a play, and label the edges of the new graph with this same recorded weight. When this simple transformation does not suffice, we mention it explicitly.

Let us first convince the reader that one can, in polynomial time, modify Inf and Sup games so that they become prefix independent.

Consider a weighted arena $\mathcal{G} = (V, V_\exists, v_I, E, w)$. We describe how to construct $\mathcal{G}_{\min}$ from $\mathcal{G}$ so that there is a clear bijection between plays in both games defined with the Inf payoff function. The arena $\mathcal{G}_{\min}$ consists of the following components:

- $V' = V \times \{w(e) \mid e \in E\}$;
- $V'_\exists = \{(v, n) \in V' \mid v \in V_\exists\}$;
- $v'_I = (v_I, W)$;
- $E' \ni \big((u, n), (v, m)\big)$ if and only if $(u, v) \in E$ and $m = \min\{n, w(u, v)\}$;
- $w'\big((u, n), (v, m)\big) = m$.

Intuitively, the construction keeps track of the minimal weight witnessed by a play by encoding it into the vertices themselves. It is not hard to see that plays in $\mathcal{G}_{\min}$ indeed have a one-to-one correspondence with plays in G. Furthermore, the LimInf and LimSup values of a play in $\mathcal{G}_{\min}$ are easily seen to be equivalent to the Inf value of the corresponding play in $\mathcal{G}$. A similar idea can be used to construct weighted arena $\mathcal{G}_{\max}$ from a Sup game such that the maximal weight is recorded (instead of the minimal). The following result then follows from the above arguments.

Lemma 4.1. *For a given weighted arena $\mathcal{G}$, and payoff function Sup: $\mathbf{Reg}(G) = \mathbf{Reg}(\mathcal{G}_{\max})$; for payoff function Inf: $\mathbf{Reg}(G) = \mathbf{Reg}(\mathcal{G}_{\min})$.* ∎

In fact, the antagonistic and co-operative values are also preserved by this transformation.

Note that both $\mathcal{G}_{\max}$ and $\mathcal{G}_{\min}$ have size linear with respect to the size of $\mathcal{G}$. Henceforth we will thus consider Sup and Inf as being prefix independent.

4.4 Contributions

In Chapter 5 we give algorithms to compute regret-minimizing strategies for Eve when playing against an arbitrary adversary. There, we give upper bounds for the associated decision problem: does Eve have a strategy to ensure her regret is of at most a given threshold. In Chapter 6 we turn our attention to the case when Adam is only allowed to use memoryless strategies. Finally, in Chapter 7 we establish the connection between minimizing regret against an oblivious adversary and determinization of quantitative automata. We then study the same problems as before, in this context. Our results are summarized in Table 4.2.

All chapters in the first part of this document are based on two articles. The first one, corresponding to our study of prefix-independent quantitative games, was first presented at the 2015 *International Conference on Concurrency Theory* and a later extended version appeared in *Acta Informatica.* The results regarding discounted-sum games have been presented at the 2016 *EACSL Annual Conference on Computer Science.*

Payoff fun.	Any strat.	Pos. strats.	Word strats.	Word strats. threshold 0
Inf, Sup, LimSup	PTIME-c (Thm 5.1, 5.3)	PSPACE (Thm 6.3), coNP-h (Thm 6.1)	EXPTIME-c (Thm 7.6)	NP-c (Thm 7.1, 7.5)
LimInf	PTIME-c (Thm 5.3)	PSPACE-c (Thm 6.2, 6.3)	EXPTIME-c (Thm 7.6)	NP-c (Thm 7.1, 7.5)
$\underline{\mathsf{MP}}$, $\overline{\mathsf{MP}}$	MPG-eq. (Thm 5.1, 5.3)	PSPACE-c (Thm 6.2, 6.3)	UNDEC (Thm 7.4)	NP-c (Thm 7.1, 7.5)
DS_λ	NP (Lem 5.1)	PSPACE-c (Thm 6.2, Lem 6.12)	EXPTIME if $\lambda = \frac{1}{b}$ (Thm 7.7), PSPACE-c for ε-gap (Thm 7.2, 7.8)	NP-c (Thm 7.1, 7.5)

Table 4.2: Complexity of deciding the regret threshold problem.

Chapter 5

Minimizing Regret Against an Unrestricted Adversary

In this chapter, we provide algorithms to solve the Regret Threshold Problem: given a game and a threshold, does there exist a strategy for Eve with a regret of at most the threshold against all strategies for Adam. It is worth mentioning that we actually provide algorithms to solve the following, more general, search problem: find the controller which ensures the minimal possible regret. Our algorithms are reductions to well-known games. We establish two links between the original and the constructed game. First, a worst-case optimal strategy for Eve in the constructed game corresponds to a regret-minimizing strategy in the original one. Second, the regret value of the original game is a function of the antagonistic value of the constructed game.

We study this problem for seven common quantitative measures: Inf, Sup, LimInf, LimSup, $\underline{\mathsf{MP}}$, $\overline{\mathsf{MP}}$, DS_λ. For all measures, but MP and DS_λ, the strict and non-strict threshold problems are equivalent. We state our results for both cases for consistency. In almost all the cases, we provide matching lower bounds showing the worst-case optimality of our algorithms.

Contributions. For prefix-independent and Inf and Sup games we establish that the problem of computing the antagonistic value and the problem of computing the regret value are *inter-reducible* in polynomial time (see Theorem 5.1 and Theorem 5.2). For the discounted sum payoff function, the situation is more complicated. In Section 5.4 we present different approaches to decide both the 0-regret problem and the general r-regret threshold problem.

5.1 Additional Preliminaries for Regret

In the following chapters, we denote the set of direct successors of a vertex u as $\mathbf{succ}(u)$. Also, we apply the notation for sub-sequences of runs in automata to vertex sequences. For instance, if we let $\pi = v_0 v_1 \ldots$ be a play, then by $\pi[i..]$ we mean the suffix $v_i \ldots$

We will often establish relations between values and strategies of a given game $\mathcal{G}$ and values and strategies of a second game $\hat{\mathcal{G}}$. For convenience, and in order to improve readability, we use σ to denote a strategy for a player in $\mathcal{G}$ and

$\hat{\sigma}$ for a strategy in $\hat{\mathcal{G}}$; π to denote a play in $\mathcal{G}$ and $\hat{\pi}$ for a play in $\hat{\mathcal{G}}$; $\mathbf{Val}_{\mathcal{G}}(\sigma, \tau)$ to denote the value of the outcome of σ and τ in $\mathcal{G}$ and $\mathbf{Val}_{\hat{\mathcal{G}}}(\hat{\sigma}, \hat{\tau})$ for the value of the outcome of $\hat{\sigma}$ and $\hat{\tau}$ in $\hat{\mathcal{G}}$; etc.

In the unfolded definition of the regret of a game, *i.e.*

$$\mathbf{Reg}_{\Sigma_\exists, \Sigma_\forall}(\mathcal{G}) := \inf_{\sigma \in \Sigma_\exists} \sup_{\tau \in \Sigma_\forall} (\sup_{\sigma' \in \Sigma_\exists} \mathbf{Val}(\sigma', \tau) - \mathbf{Val}(\sigma, \tau)),$$

let us refer to the witnesses σ and σ' as the *primary strategy* and the *alternative strategy* respectively. Observe that for any primary strategy for Eve and any one strategy for Adam, we can assume Adam plays to maximize the payoff (*i.e.* co-operates) together with the alternative strategy once it deviates (necessarily at an Eve vertex). Indeed, since the deviation yields different histories, the two strategies for Adam can be combined without conflict. More formally,

Proposition 5.1. *Consider any $\sigma \in \mathfrak{S}_\exists$, $\tau \in \mathfrak{S}_\forall$, and corresponding play $\pi_{\sigma\tau} = v_0 v_1 \ldots$. For all $i \geq 0$ such that $v_i \in V_\exists$, for all $v' \in \mathbf{succ}(v_i) \setminus \{v_{i+1}\}$ there exist $\sigma' \in \mathfrak{S}_\exists$, $\tau' \in \mathfrak{S}_\forall$ for which*

(i) $\pi_{\sigma'\tau}[..i+1] = \pi_{\sigma\tau}[..i] \cdot v'$,

(ii) $\mathbf{Val}(\pi_{\sigma'\tau'}[i+1..]) = \mathbf{cVal}^{v'}(\mathcal{G})$, *and*

(iii) $\pi_{\sigma\tau} = \pi_{\sigma\tau'}$.

■

Furthermore, from any vertex $v \in V$, Eve has a strategy to ensure a payoff of at least $\mathbf{aVal}^v(\mathcal{G})$ and Adam has a strategy to ensure a payoff of at most $\mathbf{aVal}^v(\mathcal{G})$. Thus, one could further assume that Adam plays to minimize against the primary strategy while maximizing against the alternative one.

Proposition 5.2. *Consider any $\sigma \in \mathfrak{S}_\exists$, $\tau \in \mathfrak{S}_\forall$, and corresponding play $\pi_{\sigma\tau} = v_0 v_1 \ldots$. For all $i \geq 0$ such that $v_i \in V_\exists$, for all $v' \in \mathbf{succ}(v_i) \setminus \{v_{i+1}\}$ there exist $\sigma' \in \mathfrak{S}_\exists$, $\tau' \in \mathfrak{S}_\forall$ for which*

(i) $\pi_{\sigma'\tau}[..i+1] = \pi_{\sigma\tau}[..i] \cdot v' = \pi_{\sigma\tau'}[..i] \cdot v'$,

(ii) $\mathbf{Val}(\pi_{\sigma'\tau'}[i+1..]) = \mathbf{cVal}^{v'}(\mathcal{G})$, *and*

(iii) $\mathbf{Val}(\pi_{\sigma\tau'}[i+1..]) \leq \mathbf{aVal}^{v_{i+1}}(\mathcal{G})$.

■

Both claims follow from the definitions of strategies for Eve and Adam and from Proposition 3.5.

5.2 Lower Bounds

For all the payoff functions, from a given game $\mathcal{G}$ we can construct in logarithmic space $\mathcal{G}'$ such that the antagonistic value of $\mathcal{G}$ is a function of the regret value of $\mathcal{G}'$, and so we have:

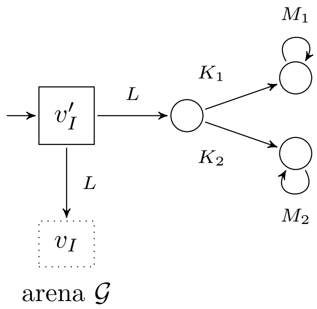

Figure 5.1: Gadget to reduce a game to its regret game.

Theorem 5.1. *For payoff functions* Inf*,* Sup*,* LimInf*,* LimSup*,* $\underline{\mathsf{MP}}$*,* $\overline{\mathsf{MP}}$*, and* DS_λ*, computing the regret value of a game is at least as hard as computing the antagonistic value of a (polynomial-size) game with the same payoff function.* ■

Suppose $\mathcal{G}$ is a weighted arena with initial vertex v_I. Consider the weighted arena $\mathcal{G}'$ obtained by adding to $\mathcal{G}$ the gadget of Figure 5.1. The initial vertex of $\mathcal{G}'$ is set to be v'_I. In $\mathcal{G}'$ from v'_I Eve can either progress to the original game or to the new gadget, both with weight L. We claim that the right choice of values for the parameters K_1, K_2, L, M_1, M_2 makes it so that the antagonistic value of $\mathcal{G}$ is a function of the regret of the game $\mathcal{G}'$.

Proof of Theorem 5.1 for prefix-independent functions. For all payoff functions that are prefix independent we set K_1 and M_1 to $w_{\max} + 1$, and K_2 and M_2 to $-3w_{\max} - 2$. For Inf we have $L = w_{\max}$, for Sup we have $L = -w_{\max}$ and for the remaining payoff functions we have $L = 0$.

Observe that for all prefix-independent payoff functions we consider we have that $\mathbf{aVal}(\mathcal{G})$ and $\mathbf{cVal}(\mathcal{G})$ both lie in $[-w_{\max}, w_{\max}]$.

Claim 1. *For payoff functions* Inf*,* Sup*,* LimInf*,* LimSup*,* $\underline{\mathsf{MP}}$*, and* $\overline{\mathsf{MP}}$ *it holds that:*

$$\mathbf{aVal}(\mathcal{G}) = w_{\max} + 1 - \mathbf{Reg}(\mathcal{G}').$$

■

Indeed, at v'_I Eve has a choice: she can choose to remain in the gadget or she can move to the original game $\mathcal{G}$. If she chooses to remain in the gadget, her payoff will be $-3w_{\max} - 2$, meanwhile Adam could choose a strategy that would have achieved a payoff of $\mathbf{cVal}(\mathcal{G})$ if she had chosen to play to $\mathcal{G}$. Hence her regret in this case is $\mathbf{cVal}(\mathcal{G}) + 3w_{\max} + 2 \geq 2w_{\max} + 2$. Otherwise, if she chooses to play to $\mathcal{G}$ she can achieve a payoff of at most $\mathbf{aVal}(\mathcal{G})$. As $\mathbf{cVal}(\mathcal{G}) \leq w_{\max}$ is the maximum possible payoff achievable in $\mathcal{G}$, the strategy which now maximizes the regret of Eve is the one which remains in the gadget—giving a payoff of $w_{\max}+1$. Her regret in this case is $w_{\max}+1-\mathbf{aVal}(\mathcal{G}) \leq 2w_{\max}+1$. Therefore, to minimize her regret she will play this strategy, and $\mathbf{Reg}(\mathcal{G}') = w_{\max} + 1 - \mathbf{aVal}(\mathcal{G})$.

The desired result follows from the claim. □

A different valuation of the parameters in $\mathcal{G}'$ yields the proof for the remaining payoff function.

Proof of Theorem 5.1 for discounted sum. Let K_1 be assigned the value $\frac{w_{\max}}{1-\lambda} + 1$, $K_2 = -3\left(\frac{w_{\max}}{1-\lambda}\right) - 2$, and $M_1 = M_2 = L = 0$. Note that, for the discounted sum function, we have that $-\frac{w_{\max}}{1-\lambda} \leq \mathbf{aVal}(\mathcal{G}) \leq \mathbf{cVal}(\mathcal{G}) \leq \frac{w_{\max}}{1-\lambda}$. Using the same arguments as for prefix-independent functions, it is then easy to show that

Claim 2. *For payoff function* DS_λ *it holds that*

$$\mathbf{aVal}(\mathcal{G}) = \frac{w_{\max}}{1-\lambda} + 1 - \mathbf{Reg}(\mathcal{G}')/\lambda.$$

■

The desired result once more follows from the claim. □

5.3 Upper Bounds for Prefix-Independent Functions

In this section we provide an algorithm to decide the regret threshold problem for games with prefix-independent objectives. Our solution consists in constructing a new game with the same payoff function. The regret value of the original game can then be computed as a function of the antagonistic value of the new game. The algorithm is thus optimal in view of the lower bound provided in Section 5.2.

Theorem 5.2. *For payoff functions* Inf, Sup, LimInf, LimSup, $\underline{\mathsf{MP}}$, *and* $\overline{\mathsf{MP}}$, *computing the regret value of a game is at most as hard as computing the antagonistic value of a (polynomial-size) game with the same payoff function.* ■

Proof. The result follows from Claim 3 for all prefix-independent payoff functions. Together with Lemma 4.1, we get the same result for Inf and Sup. □

Let us fix a weighted arena $\mathcal{G}$. We define a new weight function w' as follows. For any edge $e = (u, v)$ let $w'(e) = -\infty$ if $u \in V \setminus V_\exists$, and if $u \in V_\exists$ then

$$w'(e) = \max\{\mathbf{cVal}^{v'} \mid (u, v') \in E \setminus \{e\}\}.$$

Intuitively, w' represents the best value obtainable for a strategy of Eve that differs at the given edge. It is not difficult to see that in order to minimize regret, Eve is trying to minimize the difference between the value given by the original weight function w and the value given by w'. For $b \in \mathsf{Range}(w')$ we define $\mathcal{G}^b$ to be the graph obtained by restricting $\mathcal{G}$—the original weighted arena with weight function w—to edges e with $w'(e) \leq b$.

Next, we will construct a new weighted arena $\hat{\mathcal{G}}$ such that the regret of $\mathcal{G}$ is a function of the *antagonistic* value of $\hat{\mathcal{G}}$. Figure 5.2 depicts the general form of the arena we construct. We have three vertices $v_0 \in \hat{V} \setminus \hat{V_\exists}$ and $v_1, v_\perp \in \hat{V_\exists}$ and a "copy" of $\mathcal{G}$ as $\mathcal{G}^b$ for each $b \in \mathsf{Range}(w') \setminus \{-\infty\}$. We have a self-loop of weight 0 on v_0 which is the initial vertex of $\hat{\mathcal{G}}$, a self-loop of weight $-2w_{\max} - 1$ on $v_\perp$, and weight-0 edges from v_0 to v_1 and from v_1 to the initial vertices of $\mathcal{G}^b$ for all b. Recall that $\mathcal{G}^b$ might not be total. To fix this we add, for all vertices without

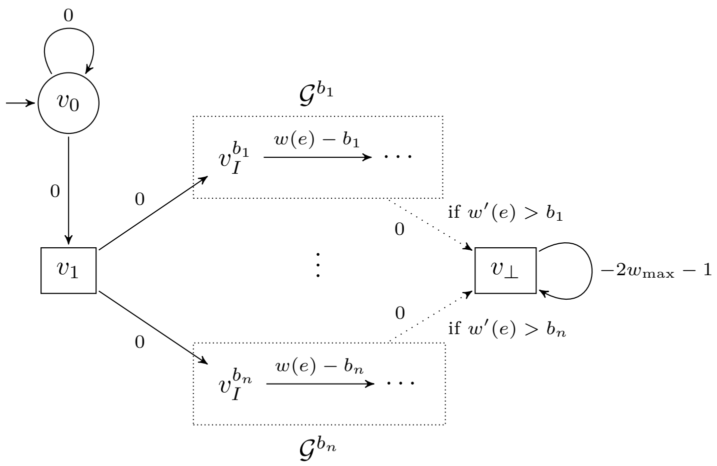

Figure 5.2: Weighted arena $\hat{\mathcal{G}}$, constructed from $\mathcal{G}$. Dotted lines represent several edges added when the condition labelling them is met.

a successor, a weight-0 edge to $v_\perp$. The remainder of the weight function $\hat{w}$, is defined for each edge e^b in $\mathcal{G}^b$ as $\hat{w}(e^b) = w(e) - b$.

Intuitively, in $\hat{\mathcal{G}}$ Adam first decides whether he can ensure a non-zero regret. If this is the case, then he moves to v_1. Next, Eve chooses a maximal value she will allow for strategies which differ from the one she will play (this is the choice of b). The play then moves to the corresponding copy of $\mathcal{G}$, *i.e.* $\mathcal{G}^b$. She can now play to maximize her payoff. However, if her choice of b was not correct then the play will end in $v_\perp$.

We show that, for all prefix-independent payoff functions we consider, the following holds:

Claim 3. *For all prefix-independent payoff functions considered in this work*

$$\mathbf{Reg}(\mathcal{G}) = -\mathbf{aVal}(\hat{\mathcal{G}}).$$

■

Proof. It follows from the definition of regret and the observation that the regret of a game is non-negative that

$$\mathbf{Reg}(\mathcal{G}) = \inf_{\sigma \in \mathfrak{S}_\exists} \sup_{\tau \in \mathfrak{S}_\forall} \sup_{\sigma' \in \mathfrak{S}_\exists \setminus \{\sigma\}} \{0, \mathbf{Val}(\sigma', \tau) - \mathbf{Val}(\sigma, \tau)\}. \tag{5.1}$$

Thus, Adam can always ensure the regret of a game is at least 0. Now, for $b \in \mathsf{Range}(w')$, define $\Sigma_\exists(b) \subseteq \mathfrak{S}_\exists(\mathcal{G})$ as:

$$\Sigma_\exists(b) := \{\sigma \mid \sup_{\tau \in \mathfrak{S}_\forall} \sup_{\sigma' \in \mathfrak{S}_\exists \setminus \{\sigma\}} \mathbf{Val}(\sigma', \tau) \leq b\}.$$

It is clear from the definitions that $\sigma \in \Sigma_\exists(b)$ if and only if σ is a strategy for Eve in $\mathcal{G}^b$ which avoids ever reaching $v_\perp$. (Note that $v_\perp$ is indeed in $\hat{\mathcal{G}}$ and not in $\mathcal{G}^b$.) Now, if we let

$$b_\sigma = \sup_{\tau \in \mathfrak{S}_\forall} \sup_{\sigma' \in \mathfrak{S}_\exists \setminus \{\sigma\}} \mathbf{Val}(\sigma', \tau),$$

then $\sigma \in \Sigma_\exists(b)$ if and only $b_\sigma \leq b$. It follows that for all σ:

$$\sup_{\tau \in \mathfrak{S}_\forall} \sup_{\sigma' \in \mathfrak{S}_\exists \setminus \{\sigma\}} \mathbf{Val}(\sigma', \tau) = \inf\{b \mid \sigma \in \Sigma_\exists(b)\}. \tag{5.2}$$

We now turn to the game played on $\hat{\mathcal{G}}$, and make some observations about the strategies we need to consider. It is well known that memoryless strategies suffice for either player to ensure an antagonistic value of at least (resp. at most) $\mathbf{aVal}(\hat{\mathcal{G}})$, for all quantitative games considered in this work, so we can assume that Adam and Eve play positionally. It follows that all plays either remain in v_0, or move to $\mathcal{G}^b$ for some b, and Adam can ensure a non-positive payoff. Note that for $b_{\max} = \max(\mathsf{Range}(w') \setminus \{-\infty\})$ we have $\mathcal{G}^{b_{\max}} = \mathcal{G}$. So the copy of $\mathcal{G}^{b_{\max}}$ in $\hat{\mathcal{G}}$ has no edge to $v_\perp$, and by playing to this sub-graph Eve can ensure a payoff of at least $-|b_{\max} - w_{\max}| \geq -2w_{\max}$. As any play that reaches $v_\perp$ will have a payoff of $-2w_{\max} - 1$, we can restrict Eve to strategies which avoid $v_\perp$, and hence all plays either remain in v_0 or (eventually) in the copy of $\mathcal{G}^b$ for some b. Now $\mathcal{G}^b$ contains no restrictions for Adam, so we can assume that he plays the same strategy in all the copies of $\mathcal{G}^b$ (where he cannot force the play to $v_\perp$), and these strategies have a one-to-one correspondence with strategies in $\mathcal{G}$. Likewise, as Eve chooses a unique $\mathcal{G}^b$ to play in, we have a one-to-one correspondence with strategies of Eve in $\hat{\mathcal{G}}$ and strategies in $\mathcal{G}$. More precisely, if $\hat{\sigma} \in \mathfrak{S}_\exists(\hat{\mathcal{G}})$ is such that $\hat{\sigma}(v_1) = v_I^b$ and $\hat{\sigma}$ avoids $v_\perp$, then the corresponding strategy $\sigma \in \mathfrak{S}_\exists(\mathcal{G})$ is a valid strategy in $\mathcal{G}^b$, and hence:

$$\hat{\sigma}(v_1) = v_I^b \implies \sigma \in \Sigma_\exists(b). \tag{5.3}$$

Now suppose $\hat{\sigma} \in \mathfrak{S}_\exists(\hat{\mathcal{G}})$ is a strategy such that $\hat{\sigma}(v_1) = v_I^b$ and $\hat{\sigma}$ avoids $v_\perp$, and $\hat{\tau} \in \mathfrak{S}_\forall(\hat{\mathcal{G}})$ is a strategy such that $\hat{\tau}(v_0) = v_1$. Let $\sigma \in \Sigma_\exists(b)$ and $\tau \in \mathfrak{S}_\forall(\mathcal{G})$ be the strategies in $\mathcal{G}$ corresponding to $\hat{\sigma}$ and $\hat{\tau}$ respectively. It is easy to show that:

$$-\mathbf{Val}_{\hat{\mathcal{G}}}(\hat{\sigma}, \hat{\tau}) = b - \mathbf{Val}_{\mathcal{G}}(\sigma, \tau). \tag{5.4}$$

Putting together Equations (5.1)–(5.4) gives:

$$\begin{aligned}
-\mathbf{aVal}(\hat{\mathcal{G}}) &= -\sup_{\hat{\sigma}} \inf_{\hat{\tau}} \mathbf{Val}_{\hat{\mathcal{G}}}(\hat{\sigma}, \hat{\tau}) \\
&= \inf_{\hat{\sigma}} \sup(\{-\mathbf{Val}_{\hat{\mathcal{G}}}(\hat{\sigma}, \hat{\tau}) \mid \hat{\tau}(v_0) = v_1\} \cup \{0\}) \\
&= \inf\{\sup(\{-\mathbf{Val}_{\hat{\mathcal{G}}}(\hat{\sigma}, \hat{\tau}) \mid \hat{\tau}(v_0) = v_1\} \cup \{0\}) \mid \hat{\sigma}(v_1) = v_I^b\} \\
&= \inf\{\sup_{\tau \in \mathfrak{S}_\forall}(\{b - \mathbf{Val}_{\mathcal{G}}(\sigma, \tau)\} \cup \{0\}) \mid \sigma \in \Sigma_\exists(b)\} \\
&= \inf_{\sigma \in \mathfrak{S}_\exists} \sup_{\tau \in \mathfrak{S}_\forall}(\{\inf\{b \mid \sigma \in \Sigma_\exists(b)\} - \mathbf{Val}_{\mathcal{G}}(\sigma, \tau)\} \cup \{0\}) \\
&= \inf_{\sigma \in \mathfrak{S}_\exists} \sup_{\tau \in \mathfrak{S}_\forall} \sup_{\sigma' \in \mathfrak{S}_\exists}\{0, \mathbf{Val}_{\mathcal{G}}(\sigma', \tau) - \mathbf{Val}_{\mathcal{G}}(\sigma, \tau)\} \\
&= \mathbf{Reg}(\mathcal{G}) \text{ as required.}
\end{aligned}$$

□

Memory requirements for Eve and Adam. It follows from the reductions underlying the proof of Theorem 5.2 that Eve only requires positional strategies to minimize regret when there is no restriction on Adam's strategies. On the other hand, for any given strategy σ for Eve, the strategy τ for Adam which

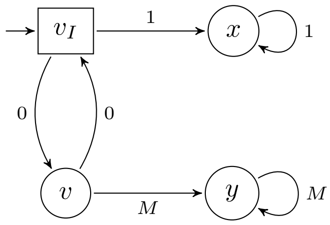

Figure 5.3: A game in which waiting is required to minimize regret.

witnesses the maximal regret against it consists of a combination of three positional strategies: first he moves to the optimal vertex for deviating (it is from this vertex that the alternative strategy σ' of Eve will achieve a better payoff against τ), then he plays his optimal (positional) strategy in the antagonistic game (*i.e.* against σ). His strategy for the alternative scenario, *i.e.* against σ', is his optimal strategy in the co-operative game which is also positional. This combined strategy is clearly realizable as a strategy with three memory states, giving us:

Corollary 1. *For payoff functions* LimInf*,* LimSup*,* $\underline{\mathsf{MP}}$ *and* $\overline{\mathsf{MP}}$*:*

$$\mathbf{Reg}(\mathcal{G}) = \mathbf{Reg}_{\mathfrak{S}^1_\exists, \mathfrak{S}^3_\forall}(\mathcal{G}).$$

■

The algorithm we give relies on the prefix independence of the payoff function. As the transformation from Inf and Sup to equivalent prefix-independent ones is polynomial it follows that polynomial memory (w.r.t. the size of the underlying graph of the arena) suffices for both players.

5.4 Upper Bounds for Discounted Sum

In this section we describe an algorithm to compute the (minimal) regret of a discounted-sum game when there are no restrictions placed on the strategies of Adam. The algorithm can be implemented by an alternating machine guaranteed to halt in pseudo-polynomial time. We show that the regret *value* of any game is achieved by a strategy for Eve which consists of two strategies, the first choosing edges which lead to the optimal co-operative value, the second choosing edges which ensure the antagonistic value. The switch from the former to the latter is done based on the “local regret” of the vertex (this is formalized in the sequel). The latter allows us to claim NP-membership of the regret threshold problem when λ is not part of the input. Additionally, we give an alternative algorithm to deal with the regret threshold problem for threshold 0. We show that this case is in NP and in coNP even if λ is part of the input. Our algorithms consists in reducing the decision problem to determining the winner of a safety game.

While for mean-payoff objectives, strategies that minimize regret are memoryless when Adam can play any strategy, we show here that pseudo-polynomial memory is necessary (and sufficient) to minimize regret in games with the discounted-sum payoff function. The need for memory is illustrated by the following example.

Example 5. Consider the example in Figure 5.3 where $M \gg 1$. Eve can play the following strategies in this game: let $i \in \mathbb{N} \cup \{\infty\}$, and note σ^i the strategy that first plays i rounds the edge (v_I, v) and then switches to (v_I, x). The regret values associated to those strategies are as follows. The regret of σ^∞ is $\frac{1}{1-\lambda}$ and it is witnessed when Adam never plays the edge (v, y). Indeed, the discounted sum of the outcome in that case is 0, while if Eve had chosen to play (v_I, x) at the first step instead, then she would have gained $\frac{1}{1-\lambda}$. The regret of σ^i is equal to the maximum between $\frac{1}{1-\lambda} - \lambda^{2i}\frac{1}{1-\lambda}$ and $\lambda^{2i+1}\frac{M}{1-\lambda} - \lambda^{2i}\frac{1}{1-\lambda}$. The maximum is either witnessed when Adam never plays (v, y) or plays (v, y) if the edge (v_I, x) has been chosen $i + 1$ times (one more time compared to σ^i). So the strategy that minimizes regret is the strategy σ^N for $N > \frac{-\log M}{2\log\lambda} - \frac{1}{2}$ (so that $\lambda^{2N+1}M < 1$), *i.e.* the strategy needs to count up to N. ■

The following theorem summarizes the bounds we obtain:

Theorem 5.3. *Deciding if the regret value of a discounted-sum game is less than a given threshold r (strictly or non-strictly), playing against all strategies of Adam, is in* EXPSPACE*; in* NP*, if λ is not part of the input; in* NP *and in* coNP*, if $r = 0$.* ■

Proof. The result follows from Lemma 5.1, Lemma 5.5, and Lemma 5.6. □

Let us start by formalizing the concept of *local regret.* Given a play or play prefix $\pi = v_0 \dots$ and integer $0 \le i < |\pi|$ such that $v_i \in V_\exists$, define $\mathbf{locreg}(\pi, i)$ as follows:

$$\begin{cases} \lambda^i \left(\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G}) - \mathbf{Val}(\pi[i..])\right) & \text{if } \pi \text{ is a play,} \\ \lambda^i \left(\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G}) - \mathbf{Val}(\pi[i..j])\right) - \lambda^j \mathbf{aVal}^{v_j}(\mathcal{G}) & \text{if } \pi \text{ is a prefix of} \\ & \text{length } j+1 > i+1, \\ \lambda^i \left(\mathbf{cVal}^{v_i}(\mathcal{G}) - \mathbf{aVal}^{v_i}(\mathcal{G})\right) & \text{if } \pi \text{ is a prefix of} \\ & \text{length } i+1, \end{cases}$$

where $\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G}) = \max\{w(v_i, v) + \lambda \mathbf{cVal}^v(\mathcal{G}) \mid (v_i, v) \in E \text{ and } v \neq v_{i+1}\}$. Intuitively, for π a play, $\mathbf{locreg}(\pi, i)$ corresponds to the difference between the value of the best *deviation* from position i and the value of π. For π a play prefix, $\mathbf{locreg}(\pi, i)$ assumes that after position $j = |\pi| - 1$ Eve will play a worst-case optimal strategy.

5.4.1 Deciding 0-regret

We will now argue that the problem of determining whether Eve has a regret-free strategy can be decided in pseudo-polynomial time and the the corresponding decision problem is in NP ∩ coNP. Furthermore, if no such strategy for Eve exists, we will extract a strategy for Adam which, against any strategy of Eve, ensures non-zero regret. To do so, we will reduce the problem to that of deciding whether Eve wins a safety game. The unsafe edges are determined by a function of the antagonistic and co-operative values of the original game. Critically, the game is played on the same arena as the original regret game.

Lemma 5.1. *Deciding if the regret value is* 0*, playing against all strategies of Adam, is in* NP ∩ coNP. ■

Proof. We define a partition of the edges leaving vertices from $V_\exists$ into good and bad for Eve. A bad edge is one which witnesses non-zero local regret. We then show that Eve can ensure a regret value of 0 if and only if she has a strategy to avoid ever traversing bad edges. More formally, let us assume a given weighted arena $\mathcal{G} = (V, V_\exists, v_I, E, w)$ and a discount factor $\lambda \in (0, 1)$. We define the set of bad edges $\mathcal{B} := \{(u, v) \in E \mid u \in V_\exists \text{ and } w(u, v) + \lambda \mathbf{aVal}^v(\mathcal{G}) < \mathbf{cVal}^u_{\neg v}(\mathcal{G})\}$.

Note that strategies for either player in the newly defined safety game are also strategies for them in the original game (and vice versa as well). We now claim that winning strategies for Adam in the safety game $\hat{\mathcal{G}} = (V, V_\exists, v_I, E, \mathcal{B})$ ensure that, regardless of the strategy of Eve, its regret will be strictly positive. The idea behind the claim is that, Adam can force to traverse a bad edge and from there, play adversarially against the primary strategy and co-operatively with an alternative strategy.

Claim 4. *If $\tau \in \mathfrak{S}_\forall$ is a winning strategy for Adam in $\hat{\mathcal{G}}$, then there exist $\tau' \in \mathfrak{S}_\forall$ and $\sigma' \in \mathfrak{S}_\exists$ such that $\forall \sigma \in \mathfrak{S}_\exists : \mathbf{Val}(\sigma', \tau') - \mathbf{Val}(\sigma, \tau') \geq \lambda^{|V|} \min\{\mathbf{cVal}^u_{\neg v}(\mathcal{G}) - w(u, v) - \lambda \mathbf{aVal}^v(\mathcal{G}) \mid (u, v) \in \mathcal{B} \text{ and } u \in V_\exists\} > 0$.* ■

The claim follows from the definitions and Proposition 5.2. Conversely, winning strategies for Eve in $\hat{\mathcal{G}}$ are actually regret-free.

Claim 5. *If $\sigma \in \mathfrak{S}_\exists$ is a winning strategy for Eve in $\hat{\mathcal{G}}$, then $\mathbf{reg}^\sigma(\mathcal{G}) = 0$.* ■

Our argument to prove this claim requires we first show that a winning strategy for Eve ensures the antagonistic value of $\mathcal{G}$ from v_I.

The desired result then follows from Proposition 3.3 and from the fact that membership of an edge in B can be decided by computing **cVal** and a threshold query regarding **aVal**, thus in NP ∩ coNP and in pseudo-polynomial time. □

We now present a full proof of Claim 5.

Proof of Claim 5. As a first step towards proving the result, we first make the observation that any winning strategy of Eve in $\hat{\mathcal{G}}$ also ensures a value of at least $\mathbf{aVal}(\mathcal{G})$ in the discounted-sum game played on $\mathcal{G}$. More formally,

Claim 6. *If $\sigma \in \mathfrak{S}_\exists$ is a winning strategy for Eve in $\hat{\mathcal{G}}$, then*

$$\forall \tau \in \mathfrak{S}_\forall, \forall i \geq 0 : \mathbf{Val}(\pi_{\sigma\tau}[i..] = v_i \dots) \geq \mathbf{aVal}^{v_i}(\mathcal{G}). \tag{5.5}$$

■

Proof. Consider a winning strategy $\sigma \in \mathfrak{S}_\exists$ for Eve in $\hat{\mathcal{G}}$. Since safety games are positionally determined we can assume w.l.o.g. that σ is memoryless.

To convince the reader that σ has the property from Equation (5.5), we consider the product of $\mathcal{G}$ and σ—that is, the product of $\mathcal{G}$ and the finite Mealy machine realizing σ. As σ is memoryless, then this product, which we denote in the sequel by $\mathcal{G} \times \sigma$, is finite. Now, towards a contradiction, suppose that Equation (5.5) does not hold for σ. Further, let us consider an alternative (memoryless) strategy σ' of Eve which ensures $\mathbf{aVal}^v(\mathcal{G})$ from all $v \in V$. The latter exists by definition of $\mathbf{aVal}(\mathcal{G})$ and memoryless determinacy of discounted-sum games (see, *e.g.* [ZP96]).

Let $\mathcal{H}$ denote a copy of $\mathcal{G} \times \sigma$ where all edges induced by E from $\mathcal{G}$ are added—not just the ones allowed by σ—and $\mathcal{H}\restriction\sigma'$ denote the sub-graph of $\mathcal{H}$ where only edges allowed by σ' are left. Since, by assumption, σ does not have the property of Equation (5.5) then the edges present in at least one vertex from $\mathcal{H}\restriction\sigma'$ and $\mathcal{G} \times \sigma$ differ. Note that such a vertex u is necessarily such that $u \in V_\exists$. Furthermore, from our definition of a strategy, we know that there is a single outgoing edge from it in both structures. Let us write (u, v) for the edge in $\mathcal{G} \times \sigma$ and (u, v') for the edge in $\mathcal{H}\restriction\sigma'$. Recall that σ is winning for Eve in $\hat{\mathcal{G}}$. Thus, we have that $(u, v) \not\in B = \{(u, v) \in E \mid u \in V_\exists$ and $w(u, v) + \lambda\mathbf{aVal}^v(\mathcal{G}) < \mathbf{cVal}^u_{\neg v}(\mathcal{G})\}$. It follows that

$$\begin{aligned} w(u,v) + \lambda\mathbf{aVal}^v(\mathcal{H}) &\geq \max_{x \neq v}\{w(u,x) + \lambda\mathbf{cVal}^x(\mathcal{H})\} \\ &\geq \max_{x \neq v}\{w(u,x) + \lambda\mathbf{aVal}^x(\mathcal{H})\} \qquad \text{as } \mathbf{cVal} \geq \mathbf{aVal} \\ &= \mathbf{aVal}^u(\mathcal{H}) \qquad \text{because } u \in V_\exists. \end{aligned}$$

Thus, the strategy σ'' of Eve which takes (u, v) instead of (u, v') and follows σ' otherwise—indeed, this might mean σ'' is not memoryless—also achieves at least $\mathbf{aVal}^u(\mathcal{H})$ from u onwards and is therefore an worst-case optimal antagonistic strategy in $\mathcal{G}$ (*i.e.* it has the property of Equation (5.5)). Notice that this process can be repeated for all vertices in which the two structures differ. Further, since both are finite, it will eventually terminate and yield a strategy of Eve which plays exactly as σ and for which Equation (5.5) holds, which is absurd. □

Once more, consider a winning strategy $\sigma \in \mathfrak{S}_\exists$ for Eve in $\hat{\mathcal{G}}$. We will now show that

$$\forall \tau \in \mathfrak{S}_\forall, \forall \sigma' \in \mathfrak{S}_\exists \setminus \{\sigma\} : \mathbf{Val}(\sigma, \tau) \geq \mathbf{Val}(\sigma', \tau).$$

The desired result will then directly follow.

Consider arbitrary strategies $\tau \in \mathfrak{S}_\forall$ and $\sigma' \in \mathfrak{S}_\exists \setminus \{\sigma\}$. Suppose that $\pi_{\sigma\tau} \neq \pi_{\sigma'\tau}$, as our claim trivially holds otherwise. Let ι be the maximal index $i \geq 0$ such that, if we write $\pi_{\sigma\tau} = v_0 v_1 \ldots$ and $\pi_{\sigma'\tau} = v'_0 v'_1 \ldots$, then $v_i = v'_i$. That is, ι is the maximal index for which the outcomes of σ and τ, and σ' and τ coincide. Note that v_ι is necessarily an Eve vertex, *i.e.* $v_\iota \in V_\exists$. We observe that, by definition of $\mathbf{cVal}$, it holds that

$$\mathbf{Val}(\pi_{\sigma'\tau}[\iota + 1..]) \leq \mathbf{cVal}^{v'_{\iota+1}}(\mathcal{G}). \tag{5.6}$$

Furthermore, we know from the fact that σ is winning for Eve in $\hat{\mathcal{G}}$ that the edge $(v_\iota, v_{\iota+1})$ is such that

$$w(v_\iota, v_{\iota+1}) + \lambda\mathbf{aVal}^{v_{\iota+1}}(\mathcal{G}) \geq \max_{t \neq v_{\iota+1}}\{w(v_\iota, t) + \lambda\mathbf{cVal}^t(\mathcal{G})\}. \tag{5.7}$$

In particular, this implies that $w(v_\iota, v_{\iota+1}) + \lambda\mathbf{aVal}^{v_{\iota+1}}(\mathcal{G}) \geq w(v_\iota, v'_{\iota+1}) + \lambda\mathbf{cVal}^{v'_{\iota+1}}(\mathcal{G})$. It is then easy to verify that $w(v_\iota, v_{\iota+1}) + \lambda\mathbf{aVal}^{v_{\iota+1}}(\mathcal{G}) = \mathbf{aVal}^{v_\iota}(\mathcal{G})$ using the observation that $v_\iota \in V_\exists$. From Claim 6 we also get that

$$\mathbf{Val}(\pi_{\sigma\tau}[\iota..]) \geq \mathbf{aVal}^{v_\iota}(\mathcal{G}). \tag{5.8}$$

Putting all the above inequalities together, we have

$$\begin{aligned} \mathbf{Val}(\pi_{\sigma\tau}[\iota..]) &\geq \mathbf{aVal}^{v_\iota}(\mathcal{G}) = w(v_\iota, v_{\iota+1}) + \lambda\mathbf{aVal}^{v_{\iota+1}}(\mathcal{G}) \qquad \text{by Eqn. (5.8)} \\ &\geq w(v_\iota, v'_{\iota+1}) + \lambda\mathbf{cVal}^{v'_{\iota+1}}(\mathcal{G}) \qquad \text{by Eqn. (5.7)} \\ &\geq \mathbf{Val}(\pi_{\sigma'\tau}[\iota..]) \qquad \text{by Eqn. (5.6)} \end{aligned}$$

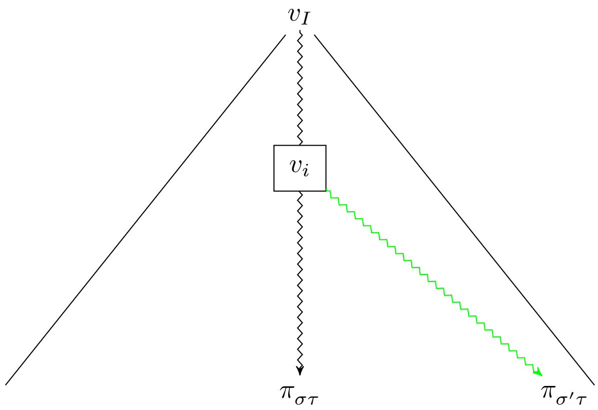

Figure 5.4: Depiction of a play and a "better alternative play".

which, in turn, implies $\mathbf{Val}(\sigma, \tau) \geq \mathbf{Val}(\sigma', \tau)$ since $\pi_{\sigma\tau}[..\iota] = \pi_{\sigma'\tau}[..\iota]$. □

We observe that the proof of Lemma 5.1—more precisely, Claim 4—implies that, if there is no regret-free strategy for Eve in a game, then the regret of the game is at least $\lambda^{|V|}$ times the smallest local regret labelling the bad edge from $\mathcal{B}$ which Adam can force. More formally:

Corollary 2. *If no regret-free strategy for Eve exists in $\mathcal{G}$, then $\mathbf{Reg}(\mathcal{G}) \geq a_{\mathcal{G}}$ where $a_{\mathcal{G}} := \lambda^{|V|} \min\{\mathbf{locreg}(uv, 0) \mid u \in V_\exists$ and $(u, v) \in B\}$.* ■

5.4.2 Deciding r-regret

It will be useful in the sequel to define the *regret of a play* and the *regret of a play prefix*. Given a play $\pi = v_0 v_1 \dots$, we define the regret of π as:

$$\mathbf{reg}(\pi) := \sup\left(\{\mathbf{locreg}(\pi, i) \mid v_i \in V_\exists\} \cup \{0\}\right).$$

Intuitively, the local regrets give lower bounds for the overall regret of a play. We will also let the *regret of a play prefix* $\varrho = v_0 \dots v_j$ be equal to

$$\max\left(\{\lambda^i(\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G}) - \mathbf{Val}(\varrho[i..j])) \mid 0 \leq i < j \text{ and } v_i \in V_\exists\} \cup \{0\}\right).$$

Let us give some more intuition regarding the regret of a play. Consider a pair of strategies σ and τ for Eve and Adam, respectively. Suppose there is an alternative strategy σ' for Eve, such that, against τ, the obtained payoff is greater than that of $\pi_{\sigma\tau}$. It should be clear that this implies there is some position i such that, from vertex $v_i \in V_\exists$ σ' and τ result in a different play from $\pi_{\sigma\tau}$ (see Figure 5.4). We will sometimes refer to this deviation, *i.e.* the play $\pi_{\sigma'\tau}$, as a *better alternative* to $\pi_{\sigma\tau}$.

We can now show the regret of a strategy for Eve in fact corresponds to the supremum of the regret of plays consistent with the strategy.

Lemma 5.2. *For any strategy σ of Eve,*

$$\mathbf{reg}^\sigma(\mathcal{G}) = \sup\{\mathbf{reg}(\pi) \mid \pi \textit{ is consistent with } \sigma\}.$$

■

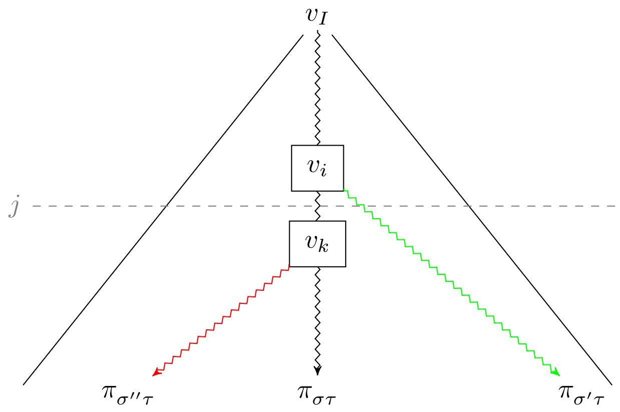

Figure 5.5: A deviation from v_k cannot be a better alternative to $\pi_{\sigma\tau}$ if $j \geq N(\mathbf{Val}(\sigma', \tau) - \mathbf{Val}(\sigma, \tau))$.

Proof. Consider any $\sigma, \sigma' \in \mathfrak{S}_\exists$ and $\tau \in \mathfrak{S}_\forall$ such that $\pi_{\sigma\tau} \neq \pi_{\sigma'\tau}$. Let us write $\pi_{\sigma\tau} = v_0 v_1 \ldots$ and $\pi_{\sigma'\tau} = v'_0 v'_1 \ldots$ and denote by ℓ the length of the longest common prefix of $\pi_{\sigma\tau}$ and $\pi_{\sigma'\tau}$. We claim that

$$\lambda^\ell\big(\mathbf{cVal}^{v_\ell}_{\neg v_{\ell+1}}(\mathcal{G}) - \mathbf{Val}(\pi_{\sigma\tau})\big) \geq \lambda^\ell\big(\mathbf{Val}(\pi_{\sigma'\tau}[\ell..]) - \mathbf{Val}(\pi_{\sigma\tau}[\ell..])\big). \quad (5.9)$$

Indeed, if we assume it is not the case, we then get that

$$\mathbf{cVal}^{v'_{\ell+1}}(\mathcal{G}) < \mathbf{Val}(\pi_{\sigma'\tau}[\ell+1..]),$$

which contradicts the definition of $\mathbf{cVal}$. Note that Proposition 5.1 actually tells us that there is another strategy τ' for Adam and a second alternative strategy σ'' for Eve which give us equality in the above equation. More formally, from Equation (5.9) and Proposition 5.1 we get that for all $\sigma \in \mathfrak{S}_\exists$, if there are $\tau \in \mathfrak{S}_\forall$ and $\sigma' \in \mathfrak{S}_\exists$ such that $\pi_{\sigma\tau} \neq \pi_{\sigma'\tau}$ then

$$\sup_{\tau,\sigma' : \pi_{\sigma\tau} \neq \pi_{\sigma'\tau}} \mathbf{Val}(\pi_{\sigma'\tau}[\ell..]) - \mathbf{Val}(\pi_{\sigma\tau}[\ell..]) = \mathbf{cVal}^{v_\ell}_{\neg v_{\ell+1}}(\mathcal{G}) - \mathbf{Val}(\pi_{\sigma\tau}). \quad (5.10)$$

We are now able to prove the result. That is, for any strategy σ for Eve:

$$\begin{aligned}
&\sup\{\mathbf{reg}(\pi) \mid \pi \text{ is consistent with } \sigma\} \\
&= \sup_{\tau \in \mathfrak{S}_\forall} \mathbf{reg}(\pi_{\sigma\tau} = v_0 v_1 \dots) && \text{def. of } \pi_{\sigma\tau} \\
&= \sup_{\tau \in \mathfrak{S}_\forall} \max \left\{ 0, \sup_{\substack{i \geq 0 \\ v_i \in V_\exists}} \lambda^i \left(\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G}) - \mathbf{Val}(\pi_{\sigma\tau}[i..]) \right) \right\} && \text{def. of } \mathbf{reg}(\cdot) \\
&= \sup_{\tau \in \mathfrak{S}_\forall} \max \left\{ 0, \sup_{\sigma' : \pi_{\sigma\tau} \neq \pi_{\sigma'\tau}} \lambda^\ell \left(\mathbf{Val}(\pi_{\sigma'\tau}[\ell..]) - \mathbf{Val}(\pi_{\sigma\tau}[\ell..]) \right) \right\} && \text{by Eq. (5.10)} \\
&= \sup_{\tau \in \mathfrak{S}_\forall} \max \left\{ 0, \sup_{\sigma' : \pi_{\sigma\tau} \neq \pi_{\sigma'\tau}} \left(\mathbf{Val}(\sigma', \tau) - \mathbf{Val}(\sigma, \tau) \right) \right\} && \text{def. of } \mathbf{Val}(\cdot), \\
& && \text{and def. of } \ell \\
&= \sup_{\tau \in \mathfrak{S}_\forall} \sup_{\sigma' \in \mathfrak{S}_\exists} \left(\mathbf{Val}(\sigma', \tau) - \mathbf{Val}(\sigma, \tau) \right) && 0 \text{ if } \pi_{\sigma\tau} = \pi_{\sigma'\tau}
\end{aligned}$$

as required. □

We note that for any play π, the sequence $\langle \lambda^i (\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G}) - \mathbf{Val}(\pi[i..])) \rangle_{i \geq 0}$ converges to 0 because $(\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G}) - \mathbf{Val}(\pi[i..]))$ is bounded by $\frac{2w_{\max}}{(1-\lambda)}$. It follows that if we have a non-zero lower bound for the regret of π, then there is some index N such that the witness for the regret occurs before N. Moreover, we can place a pseudo-polynomial upper bound on N. More precisely:

Lemma 5.3. *Let π be a play in $\mathcal{G}$ and suppose $0 < r \leq \mathbf{reg}(\pi)$. Let*

$$N(r) := \lfloor (\log r + \log(1-\lambda) - \log(2w_{\max})) / \log \lambda \rfloor + 1.$$

Then $\mathbf{reg}(\pi) = \mathbf{reg}(\pi[..N(r)]) - \lambda^{N(r)} \mathbf{Val}(\pi[N(r)..])$. ■

Proof. Observe that $N(r)$ is such that $\frac{2w_{\max}\lambda^{N(r)}}{1-\lambda} < r$. Hence, we have that for all $i \geq N(r)$ such that $v_i \in V_\exists$ it holds that

$$\lambda^i (\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G}) - \mathbf{Val}(\pi[i..])) \leq \frac{2w_{\max}\lambda^{N(r)}}{1-\lambda} < r.$$

It follows that

$$\begin{aligned}
\mathbf{reg}(\pi) &= \sup\{\lambda^i (\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G}) - \mathbf{Val}(\pi[i..])) \mid i \geq 0 \text{ and } v_i \in V_\exists\} \\
&= \max_{\substack{0 \leq i < N(r) \\ v_i \in V_\exists}} \lambda^i \left(\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G}) - \mathbf{Val}(\pi[i..N(r)]) \right) - \lambda^{N(r)} \mathbf{Val}(\pi[N(r)..])
\end{aligned}$$

as required. □

The above result gives us a bound on how far we have to unfold a game after having witnessed a non-zero lower bound, r, for the regret. If we consider the example from Figure 5.4, this translates into a bound on how many turns after v_i a deviation can still yield bigger local regret (see Figure 5.5).

Corollary 2 then gives us the required lower bound for us to be able to use Lemma 5.3.

Lemma 5.4. *If* $\mathbf{Reg}(\mathcal{G}) \geq a_{\mathcal{G}}$ *then* $\mathbf{Reg}(\mathcal{G})$ *is equal to*

$$\inf_{\sigma \in \mathfrak{S}_\exists} \sup\{\mathbf{reg}(\pi[..N(a_{\mathcal{G}})]) - \lambda^{N(a_{\mathcal{G}})} \mathbf{aVal}^{v_{N(a_{\mathcal{G}})}}(\mathcal{G}) \mid \pi = v_0 \ldots \text{ consistent with } \sigma\}.$$

■

Proof. First, note that if $\mathbf{Reg}(\mathcal{G}) > 0$ then there cannot be any regret-free strategies for Eve in $\mathcal{G}$. It then follows from Corollary 2 that $\mathbf{Reg}(\mathcal{G}) \geq a_{\mathcal{G}}$. Next, using Lemma 5.3 and the definition of the regret of a play we have that $\mathbf{Reg}(\mathcal{G})$ is equal to

$$\inf_{\sigma \in \mathfrak{S}_\exists} \sup\{\mathbf{reg}(\pi[..N(a_{\mathcal{G}})]) - \lambda^{N(a_{\mathcal{G}})} \mathbf{Val}(\pi[N(a_{\mathcal{G}})..]) \mid \pi \text{ is consistent with } \sigma\}.$$

Finally, note that it is in the interest of Eve to maximize the value

$$\lambda^{N(a_{\mathcal{G}})} \mathbf{Val}(\pi[N(a_{\mathcal{G}})..])$$

in order to minimize her regret. Conversely, Adam tries to minimize the same value. Thus, we can replace it by the antagonistic value from $\pi[N(a_{\mathcal{G}})..]$ discounted accordingly. More formally, we have

$$\begin{aligned}
&\inf_{\sigma \in \mathfrak{S}_\exists} \sup\{\mathbf{reg}(\pi[..N(a_{\mathcal{G}})]) - \lambda^{N(a_{\mathcal{G}})} \mathbf{Val}(\pi[N(a_{\mathcal{G}})..]) \mid \pi \text{ is consistent with } \sigma\} \\
= &\inf_{\sigma \in \mathfrak{S}_\exists} \sup_{\tau \in \mathfrak{S}_\forall} \mathbf{reg}(\pi_{\sigma\tau}[..N(a_{\mathcal{G}})]) - \lambda^{N(a_{\mathcal{G}})} \mathbf{Val}(\pi_{\sigma\tau}[N(a_{\mathcal{G}})..]) \\
= &\inf_{\substack{\sigma \in \mathfrak{S}_\exists \\ \sigma' \in \mathfrak{S}_\exists}} \sup_{\substack{\tau \in \mathfrak{S}_\forall \\ \tau' \in \mathfrak{S}_\forall}} \mathbf{reg}(\pi_{\sigma\tau}[..N(a_{\mathcal{G}})] = \ldots v) - \lambda^{N(a_{\mathcal{G}})} \mathbf{Val}^v(\sigma', \tau') \\
= &\inf_{\sigma \in \mathfrak{S}_\exists} \sup_{\tau \in \mathfrak{S}_\forall} \mathbf{reg}(\pi_{\sigma\tau}[..N(a_{\mathcal{G}})] = \ldots v) + \inf_{\sigma' \in \mathfrak{S}_\exists} \sup_{\tau' \in \mathfrak{S}_\forall} \left(-\lambda^{N(a_{\mathcal{G}})} \mathbf{Val}^v(\sigma', \tau')\right) \\
= &\inf_{\sigma \in \mathfrak{S}_\exists} \sup_{\tau \in \mathfrak{S}_\forall} \mathbf{reg}(\pi_{\sigma\tau}[..N(a_{\mathcal{G}})] = \ldots v) - \lambda^{N(a_{\mathcal{G}})} \left(\sup_{\sigma' \in \mathfrak{S}_\exists} \inf_{\tau' \in \mathfrak{S}_\forall} \mathbf{Val}^v(\sigma', \tau')\right) \\
= &\inf_{\sigma \in \mathfrak{S}_\exists} \sup_{\tau \in \mathfrak{S}_\forall} \mathbf{reg}(\pi_{\sigma\tau}[..N(a_{\mathcal{G}})] = \ldots v) - \lambda^{N(a_{\mathcal{G}})} \mathbf{aVal}^v(\mathcal{G})
\end{aligned}$$

as required. □

This already implies we can compute the regret value in alternating exponential time (or equivalently, exponential space [CKS81]).

Lemma 5.5. *The regret value is computable in exponential space.* ■

Proof. We first label the arena with the antagonistic and co-operative values and solve the safety game described for Lemma 5.1. The latter can be done in pseudo-polynomial time. If Eve wins the safety game $\hat{\mathcal{G}}$—as constructed in subsection 5.4.1—the regret value is 0. Otherwise, we know $a_{\mathcal{G}} > 0$ is a lower bound for the regret value. We now simulate $\mathcal{G}$ using an alternating Turing machine which halts in at most $N(a_{\mathcal{G}})$ steps. That is, a pseudo-polynomial number of steps. The simulated play prefix is then assigned a regret value as per Lemma 5.4 (recall we have already pre-computed the antagonistic value of every vertex). □

As a Corollary, the same arguments above imply that the regret value can be computed in alternating polynomial time if λ is not given as part of the input. Hence we get that

Corollary 3. *If λ is not given as part of the input, the regret value is computable in polynomial space.* ∎

Also as a side-product of the algorithm described in the above proof, we get that finite memory strategies suffice for Eve to minimize her regret in a discounted-sum game.

Corollary 4. *Let $\mu := |\Delta|^{N(a_{\mathcal{G}})}$, with $N(0) = 0$. It holds that*

$$\mathbf{Reg}_{\mathfrak{S}_\exists^\mu,\mathfrak{S}_\forall}(\mathcal{G}) = \mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{S}_\forall}(\mathcal{G}).$$

∎

5.4.3 Simple regret-minimizing behaviors

We will now argue that Eve has a simple strategy which ensures regret of at most $\mathbf{Reg}(\mathcal{G})$. Her strategy will consist of "playing co-operatively" for some turns (until a high local regret has already been witnessed) and then switch to a co-operative worst-case optimal strategy.

We will now define a family of strategies which switch from co-operative behavior to antagonistic, after a specific number of turns have elapsed (in fact, enough for the discounted local regret to be less than the desired regret). Denote by σ^{sc} a strongly co-operative strategy for Eve in $\mathcal{G}$ and by σ^{cw} a co-operative worst-case optimal strategy for Eve in $\mathcal{G}$. Recall that, by Proposition 3.10, such strategies for her always exist. Finally, given a co-operative strategy σ^{sc}, a co-operative worst-case optimal strategy σ^{cw}, and $t \in \mathbb{Q}$ let us define an *optimistic-then-pessimistic strategy for Eve* $[\sigma^{\mathsf{sc}} \xrightarrow{\mathsf{t}} \sigma^{\mathsf{cw}}]$. The strategy is such that, for any play prefix $\varrho = v_0 \dots v_n$ such that $v_n \in V_\exists$

$$[\sigma^{\mathsf{sc}} \xrightarrow{\mathsf{t}} \sigma^{\mathsf{cw}}](\varrho) = \begin{cases} \sigma^{\mathsf{sc}}(\varrho) & \text{if } |\mathbf{cOpt}(v_n)| = 1 \text{ and } \mathbf{locreg}(\varrho \cdot \sigma^{\mathsf{cw}}(\varrho), n+1) > t \\ \sigma^{\mathsf{cw}}(\varrho) & \text{otherwise.} \end{cases}$$

We claim that, when we set $t = \mathbf{Reg}(\mathcal{G})$, an optimistic-then-pessimistic strategy for Eve ensures minimal regret. That is

Proposition 5.3. *Let σ^{sc} be a strongly co-operative strategy for Eve, σ^{cw} be a Eve and a co-operative worst-case optimal strategy for Eve, and $t = \mathbf{Reg}(\mathcal{G})$. The strategy $\sigma = [\sigma^{\mathsf{sc}} \xrightarrow{\mathsf{t}} \sigma^{\mathsf{cw}}]$ has the property that $\mathbf{reg}^\sigma(\mathcal{G}) = \mathbf{Reg}(\mathcal{G})$.* ∎

This is a refinement of the strategy one can obtain from applying the algorithm used to prove Lemma 5.5.[1] The latter tells us that a regret-minimizing strategy of Eve eventually switches to a worst-case optimal behavior. For vertices where, before this switch, another edge was chosen by Eve, we argue that she must have been playing a co-operative strategy. Otherwise, she could have switched sooner.

[1]In fact, our proof of Proposition 5.3 relies on Eve requiring finite memory, to minimize her regret.

Proof of Proposition 5.3. Let us start by showing that the regret of a play π is bounded (from above) by the discounted local regret from any index i, where from the i-th turn onwards Eve plays a worst-case optimal strategy. More formally:

Claim 7. *Let $\pi = v_0 v_1 \dots$ be a play. Assume there is some $i \in \mathbb{N}$ such that*

(i) $v_i \in V_\exists$;

(ii) $\mathbf{reg}(\pi) \leq \lambda^i \mathbf{reg}(\pi[i..])$; *and*

(iii) $\mathbf{aVal}^{v_j}(\mathcal{G}) = w(v_j, v_{j+1}) + \lambda \mathbf{aVal}^{v_{j+1}}(\mathcal{G})$, *for all $j \geq i$.*

It then holds that $\mathbf{reg}(\pi) \leq \lambda^i \left(\mathbf{cVal}^{v_i}(\mathcal{G}) - \mathbf{aVal}^{v_i}(\mathcal{G})\right)$. ■

Proof. If $\mathbf{reg}(\pi) = 0$ then the claim holds trivially. Hence, let us assume $\mathbf{reg}(\pi) > 0$. It follows from Lemma 5.4 and Assumption (ii) that there exists $k \geq i$ such that $v_k \in V_\exists$ and

$$\mathbf{reg}(\pi) = \lambda^k \left(\mathbf{cVal}^{v_k}_{\neg v_{k+1}}(\mathcal{G}) - w(v_k, v_{k+1}) - \lambda \mathbf{aVal}^{v_{k+1}}(\mathcal{G}) \right).$$

Observe that $\mathbf{cVal}^{v_k}(\mathcal{G}) \geq \mathbf{cVal}^{v_k}_{\neg v_{k+1}}(\mathcal{G})$, by definition, and that from Assumption (iii) we have that $\mathbf{aVal}^{v_k}(\mathcal{G}) \leq w(v_k, v_{k+1}) + \lambda \mathbf{aVal}^{v_{k+1}}(\mathcal{G})$. Thus, we get that $\mathbf{reg}(\pi) \leq \lambda^k \left(\mathbf{cVal}^{v_k}(\mathcal{G}) - \mathbf{aVal}^{v_k}(\mathcal{G})\right)$. Also, note that by definition of $\mathbf{cVal}$ we have that

$$\mathbf{cVal}^{v_j}(\mathcal{G}) \geq w(v_j, v_{j+1}) + \lambda \mathbf{cVal}^{v_{j+1}}(\mathcal{G})$$

for all $j \geq 0$. It thus follows from Assumption (iii) and the previous arguments that $\mathbf{reg}(\pi) \leq \lambda^i \left(\mathbf{cVal}^{v_i}(\mathcal{G}) - \mathbf{aVal}^{v_i}(\mathcal{G})\right)$ as required. □

We are now ready to prove the Proposition holds.

The zero case. If $\mathbf{Reg}(\mathcal{G}) = 0$, then it follows from our reduction to safety games that Eve has a co-operative worst-case optimal strategy which minimizes regret. Indeed, it is straightforward to show that the strategy for Eve obtained from the safety game does not only ensure at least the antagonistic value, but it is also co-operative worst-case optimal. Thus, since $[\sigma^{\mathsf{sc}} \xrightarrow{0} \sigma^{\mathsf{cw}}]$ is clearly equivalent to σ^{cw} in this case, the result follows.

Non-zero regret. Assume $\mathbf{Reg}(\mathcal{G}) > 0$. It then follows from Lemma 5.4 that Eve has a finite memory strategy σ which ensures regret of at most $\mathbf{Reg}(\mathcal{G})$ (see Corollary 4) and which, furthermore, can be assumed to switch after turn $N(a_\mathcal{G})$ to a co-operative worst-case optimal strategy σ^{cw} for Eve (since such a strategy ensures at least the antagonistic value of the vertex from which Eve starts playing it). We will further assume, w.l.o.g., that for all play prefixes $\pi = v_0 \dots v_n$ with $n \leq N(a_\mathcal{G})$, $v_n \in V_\exists$ and having $\sigma^{\mathsf{cw}}(\pi) \neq \sigma^{\mathsf{sc}}(\pi) = \sigma(\pi)$, if σ switches to σ^{cw} from π onwards—that is, for all prefixes extending π—then the regret of the resulting strategy is strictly greater than $\mathbf{Reg}(\mathcal{G})$. Otherwise, one can consider the strategy resulting from the previously described switch instead of σ.

We will now argue that for all play prefixes $\pi = v_0 \dots v_n$ with $n \leq N(a_{\mathcal{G}})$ and $v_n \in V_\exists$, if $\sigma(\pi) \neq \sigma^{\mathsf{cw}}$ then $\mathbf{cOpt}(v_n)$ is a singleton and

$$\mathbf{locreg}(\pi[..n] \cdot \sigma^{\mathsf{cw}}(\pi[..n]), n+1) > \mathbf{Reg}(\mathcal{G}).$$

The desired result will follow since, in order for our assumption of $\mathbf{reg}(\sigma) = \mathbf{Reg}(\mathcal{G})$ to be true Eve must then choose the unique edge leading to the single element in $\mathbf{cOpt}(v_n)$.

Let us consider two cases.

First, if $\mathbf{locreg}(\pi[..n] \cdot \sigma^{\mathsf{cw}}(\pi[..n]), n+1) \leq \mathbf{Reg}(\mathcal{G})$, we can switch to σ^{cw} fron $\pi[..n]$ onwards. Contradicting our initial assumption.

Second, if $|\mathbf{cOpt}(v_n)| > 1$ and $\mathbf{locreg}(\pi[..n] \cdot \sigma^{\mathsf{cw}}(\pi[..n]), n+1) > \mathbf{Reg}(\mathcal{G})$, then by Claim 7 we get that the regret of the play (if we switched to σ^{cw}) is bounded above by $\lambda^n \left(\mathbf{cVal}^{v_n}(\mathcal{G}) - \mathbf{aVal}^{v_n}(\mathcal{G})\right)$. Also, since $\mathbf{cOpt}(v_n)$ is not a singleton, if Eve does not switch, then she cannot ensure a local regret of less than $\lambda^n \left(\mathbf{cVal}^{v_n}(\mathcal{G}) - \mathbf{aVal}^{v_n}(\mathcal{G})\right)$—particularly, not even by taking an edge leading to a vertex in $\mathbf{cOpt}(v_n)$. This contradicts the assumption that that switching to σ^{cw} yields strictly more regret. □

Let us now suppose that λ is not given as part of the input. We will argue that we can then decide the regret threshold problem in non-deterministic polynomial time.

Lemma 5.6. *If λ is not given as part of the input, the regret threshold problem is in* NP. ■

Proof. Indeed, we have shown the regret value can be computed using an algorithm which requires polynomial space only. This algorithm is based on a polynomial-length unfolding of the game and from it we can deduce that the regret value is representable using a polynomial number of bits. (All exponents occurring in the formula from Lemma 5.4 will be polynomial according to Lemma 5.3.) Also, we have argued that Eve has a "simple" strategy σ to ensure minimal regret. Such a strategy is defined by two polynomial-time constructible sub-strategies and the regret value of the game. Hence, it can be encoded into a polynomial number of bits itself. Furthermore, σ is guaranteed to be playing as its co-operative worst-case optimal component after $N(\mathbf{Reg}(\mathcal{G}))$ turns (see, again, Lemma 5.3), which is a polynomial number of turns. Given a regret threshold r, we claim we can verify that σ ensures regret at most r in polynomial time. This can be achieved by allowing Adam to play in $\mathcal{G}$, and against σ, with the objective of reaching an edge with high local regret before $N(\mathbf{Reg}(\mathcal{G}))$ turns. A possible formalization of this idea follows. Consider the product of $\mathcal{G}$ with a counter ranging from 1 to $N(\mathbf{Reg}(\mathcal{G}))$ where we make all vertices belong to Adam. In this game H, we make edges leaving vertices previously belonging to Eve go to a sink and define a new weight function w' which assigns to these edges their negative non-discounted local regret: going from u to v when σ dictates to go to v' yields $w(u, v') + \lambda \mathbf{aVal}^{v'}(H \times \sigma) - w(u, v) + \lambda \mathbf{cVal}^{v}(H)$. (The function w' assigns a weight of 0 to all the other edges.) Lemma 5.4 tells us that σ ensures regret at most r in $\mathcal{G}$ if and only if the antagonistic value of a discounted-sum game played on H with weight function w' is at least $-r$.

It follows that the regret threshold problem is in NP. □

We close this section by applying the algorithm we just described to an example discounted-sum game.

Example 6. We revisit the discounted-sum game from Figure 5.3. Let us instantiate the values $M = 100$ and $\lambda = \frac{9}{10}$. According to our previous remarks on this arena, after i visits to v without Adam choosing (v, y), Eve could achieve $\left(\frac{1}{1-\frac{9}{10}}\right)\left(\frac{9}{10}\right)^{2i} = 10\left(\frac{9}{10}\right)^{2i}$, by going to x, or hope for $\left(\frac{100}{1-\frac{9}{10}}\right)\left(\frac{9}{10}\right)^{2i+1} = 1000\left(\frac{9}{10}\right)^{2i+1}$ by going to v again. Her best regret minimizing strategy corresponds to σ^{22} which ensures regret of at most $10 - 10(\frac{9}{10})^{44} \approx 9.9030$. (Note that, since $1000(\frac{9}{10})^{2j+1} < 10$ for all $j \geq 22$, the best alternative strategy indeed achieves a payoff of 10 only. This alternative strategy corresponds to having gone to x from the beginning.)

It is easy to see that Eve cannot win the safety game $\hat{\mathcal{G}}$ constructed from this arena. Indeed, from the initial vertex v_I both (v_I, x) and (v_I, v) are bad edges for Eve since there are always alternative strategies to obtain higher payoffs (that is, if Adam does not play to y). More formally, we note that $\mathbf{cVal}^{v_I}_{\neg x} = 1000\lambda$, $\mathbf{cVal}^{v_I}_{\neg v} = 10$, $\mathbf{aVal}^{x} = 10$, and $\mathbf{aVal}^{v} = 10\lambda^2$. Thus, the lower bound $a_{\mathcal{G}}$ one can obtain from $\hat{\mathcal{G}}$ is then equal to

$$\begin{aligned}\min\left\{1000\left(\frac{9}{10}\right) - 10, 10 - 10\left(\frac{9}{10}\right)^2\right\}\left(\frac{9}{10}\right)^4 =&10\left(\frac{9}{10}\right)^4 - 10\left(\frac{9}{10}\right)^6 \\ \approx&1.2466.\end{aligned}$$

As expected, when Eve plays her optimal regret-minimizing (optimistic-then-pessimistic) strategy any better alternative must deviate before $N(a_{\mathcal{G}}) = 71$ turns. In general, against σ^i, for $i < 22$ a regret bigger than 9.9030 is obtained by Adam choosing the edge (v, y) to help any strategy of Eve going to v more than i times, for $i \geq 22$ choices of Adam are no longer relevant and the best alternative strategy for Eve is to have gone to x from the first step. ■

Chapter 6

Minimizing Regret Against Positional Adversaries

In this chapter, we study strategies for Eve which minimize regret against an adversary which is assumed to play positionally. Recall that all quantitative games considered in this dissertation are known to be positionally determined. That is, both Eve and Adam have optimal (maximizing and, respectively, minimizing) positional strategies. Hence, the present restriction on the set of strategies from which Adam can choose is natural.

Restricting the environment to positional behaviors is useful when designing a system that needs to perform well in an environment which is only partially known. In practical situations, a controller may discover the environment with which it is interacting at run time. Such a situation can be modelled by an arena in which choices, in nodes of the environment, model an entire family of environments and each memoryless strategy models a specific environment of the family. In such cases, if we want to design a controller that performs reasonably well against all the possible environments, we can consider a controller that minimizes regret: the strategy of the controller will be as close as possible to an optimal strategy if we had known the environment beforehand. This is, for example, the modelling choice done in the famous Canadian traveller's problem [PY91]: a driver is attempting to reach a specific location while ensuring the traversed distance is *not too far* from the shortest feasible path. The partial knowledge is due to some roads being closed because of snow. The Canadian traveller, when planning his itinerary, is in fact searching for a strategy to minimize his regret for the shortest path measure against a memoryless adversary who determines the roads that are closed. Similar situations naturally arise when synthesizing controllers for *robot motion planning* [WET15].

We will, again, consider the regret threshold problem in this setting and for the same quantitative measures as in Chapter 5: Inf, Sup, LimInf, LimSup, $\underline{\mathsf{MP}}$, $\overline{\mathsf{MP}}$, and DS_λ.

Contributions. For prefix-independent and Inf and Sup games we reduce the computation of the regret value (and, thus, the regret threshold problem) to the computation of the antagonistic value in a larger game. We then argue that, in the constructed game, the antagonistic value can be computed using

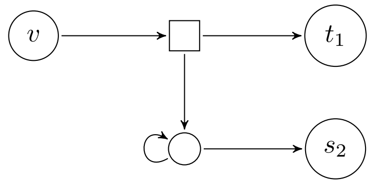

Figure 6.1: Regret gadget for 2-disjoint-paths reduction.

only polynomial space. For discounted-sum games we provide an alternative solution which follows the same ideas presented in Chapter 5. Unfortunately, the space required by the resulting algorithm is quite high, namely, exponential. We consider sub-cases for the discounted sum function and show that more efficient solutions for them exist. In particular, we study once more the special case of threshold 0 and the regret threshold problem when λ is not being given in binary as part of the input.

In terms of lower bounds, for most payoff functions, the polynomial-space algorithm we give turns out to be optimal. For Inf, Sup, LimSup, and DS_λ, however, we have not been able to show PSPACE-hardness. Nonetheless, a reduction from the problem of finding two disjoint paths in a weighted digraph, allows us to argue that they are coNP-hard (even for threshold 0).

6.1 Lower Bounds

In this section we will first show that the regret threshold problem is coNP-hard for all payoff functions we consider. We will then show an even higher lower bound, namely PSPACE, for a subset of the functions.

Theorem 6.1. *Consider a fixed regret threshold $r \in \mathbb{Q}$ and let $\lhd \in \{<, \leq\}$. For payoff functions Inf, Sup, LimInf, LimSup, $\underline{\mathsf{MP}}$, $\overline{\mathsf{MP}}$, and DS_λ, determining whether $\mathbf{Reg}_{\mathfrak{S}_\exists, \mathfrak{S}^1_\forall}(\mathcal{G}) \lhd r$ for a given weighted arena $\mathcal{G}$, is coNP-hard even if λ is not part of the input.* ■

Proof. We provide a reduction from the complement of the 2 Disjoint Paths Problem (2DP) on directed graphs [ET98]. As the problem is known to be NP-complete, the result follows. In other words, we describe how to translate a given instance of the 2DP problem into a weighted arena in which Eve can ensure regret value strictly less than r if and only if the answer to the instance of the 2DP problem is negative. We focus on payoff functions Sup and LimSup and a strict threshold for clarity. However, it will be clear how one can easily adapt the construction to all other payoff functions and the non-strict threshold problem.

Consider a directed graph $\mathcal{G}$ and distinct vertex pairs (s_1, t_1) and (s_2, t_2). W.l.o.g. we assume $\mathcal{G}$ is such that for all $i \in \{1, 2\}$:

(i) t_i is reachable from s_i, and

(ii) t_i is a sink.

We now describe the changes we apply to $\mathcal{G}$ in order to get the underlying graph structure of the weighted arena and then comment on the weight function. Let all vertices from $\mathcal{G}$ be Adam vertices and s_1 be the initial vertex. We replace all edges arriving at t_1—edges of the form (v, t_1), for some v—by a copy of the gadget shown in Figure 6.1. Next, we add self-loops on t_1 and t_2 with weights r and $2r$, respectively. Finally, the weights of all remaining edges are 0. (To make sure $\mathcal{G}$ is total, we add self-loops on all vertices with no outgoing edges.)

We claim that, in this weighted arena, Eve can ensure regret strictly less than r—for payoff functions Sup and LimSup—if and only if in $\mathcal{G}$ the vertex pairs (s_1, t_1) and (s_2, t_2) cannot be joined by vertex-disjoint paths. Indeed, we claim that the strategy that minimizes the regret of Eve is the strategy that, in states where she has a choice, tells her to go to t_1.

First, let us prove that this strategy has regret strictly less than r if and only if no two disjoint paths in the graph exist between the pairs of states (s_1, t_1) and (s_2, t_2). Assume the latter is the case. Then if Adam chooses to always avoid t_1, then clearly the regret is 0. If t_1 is eventually reached, then the choice of Eve secures a value of r (for all payoff functions). Note that if she had chosen to go towards s_2 instead, as there are no two disjoint paths, we know that either the path constructed from s_2 by Adam never reaches t_2, and then the value of the path is 0—and the regret is 0 for Eve—or the path constructed from s_2 reaches t_1 again since Adam is playing positionally—and, again, the regret is 0 for Eve. Now assume that two disjoint paths between the source-target pairs do exist. If Eve changed her strategy to go towards s_2 (instead of choosing t_1) then Adam has a strategy to reach t_2 and achieve a payoff of $2r$. Thus, her regret is at least r.

Second, we claim that any other strategy of Eve has a regret greater than or equal to r. Indeed, if Eve decides to go towards s_2 (instead of choosing to go to t_1) then Adam can choose to loop on the state before s_2 and the payoff in this case is 0. Since she could have gotten r by going to t_1, the regret of Eve is at least r.

Discounted sum. For the discounted-sum function, we weight the self-loops on t_1 and t_2 with A and B, respectively. The reduction then works, as is, for any value of A and B such that

(i) $\lambda^{|V|}\frac{A}{1-\lambda} > r$, and

(ii) $\lambda^{|V|}\frac{B}{1-\lambda} - \lambda\frac{A}{1-\lambda} > r$.

For instance, consider $\alpha := \frac{r+1}{\lambda^{|V|}}$. It is easy to verify that setting $A := (1-\lambda)\alpha$ and $B := (1-\lambda)\alpha^2$ satisfies the inequalities. Furthermore, A and B are rational numbers which can be represented using a polynomial number of bits w.r.t. $|V|$ and the size of the representation of both λ and r. □

We will now argue that minimizing regret in games where the payoff function is LimInf, mean-payoff, or discounted sum is even harder. To do so, we will reduce the QBF problem to the regret threshold problem for games with those payoff functions.

Theorem 6.2. *Let $\lhd \in \{<, \leq\}$. For payoff functions* LimInf*,* $\underline{\mathsf{MP}}$*,* $\overline{\mathsf{MP}}$*, and* DS_λ*, determining whether* $\mathbf{Reg}_{\mathfrak{S}_\exists, \mathfrak{S}^1_\forall}(\mathcal{G}) \lhd r$ *for a given weighted arena* $\mathcal{G}$ *and regret threshold* $r \in \mathbb{Q}$*, is* PSPACE*-hard.* ■

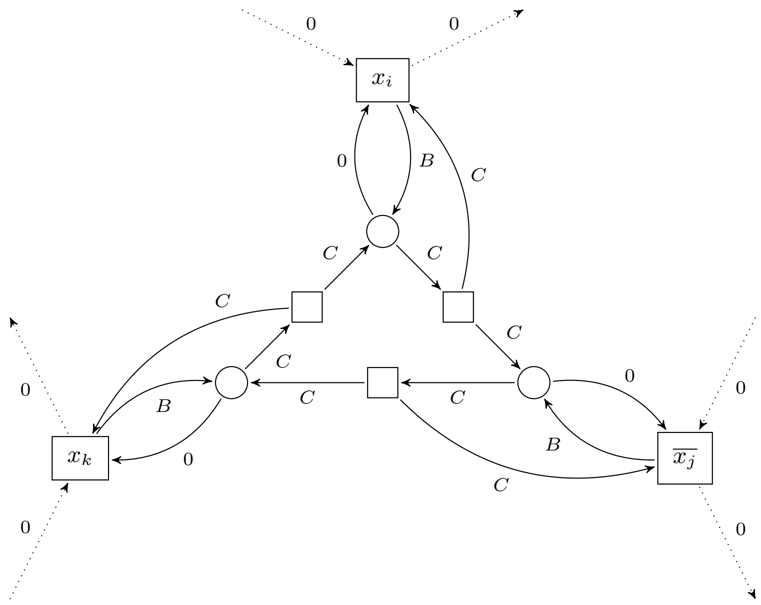

Figure 6.2: Clause gadget for the QBF reduction for clause $x_i \vee \neg x_j \vee x_k$.

Consider an instance of the QBF problem given in the following form:

$$\exists x_0 \forall x_1 \exists x_2 \ldots \Phi(x_0, x_1, \ldots, x_m)$$

where Φ is in 3-CNF and has n clauses. Also w.l.o.g., we assume that every non-trivially true clause has at least one existentially quantified variable (as otherwise the answer to the problem is trivial).

Recall that the QBF problem can be viewed as a game between an existential and a universal player. The existential player chooses a truth value for existentially quantified variable x_i and the universal player responds with a truth value for x_{i+1}. After m turns the truth value of Φ determines the winner: the existential player wins if Φ is true and the universal player wins otherwise. The game we shall construct mimics the choices of the existential and universal player and makes sure that the regret of the game is small if and only if Φ is true.

The crux of the reduction from the QBF problem is a gadget used for each clause of the QBF formula. Visiting this gadget allows Eve to gain information about the highest payoff obtainable in the gadget: each entry point corresponds to a literal from the clause, and the literal is 'visited' when it is made true by the valuation of variables chosen by Eve and Adam in the reduction described below. Figure 6.2 depicts an instance of the gadget for a particular clause. As an example, let us focus on the mean-payoff function and let $A = 4$, $B = 3$. Note that staying in the inner 6-vertex triangle would yield a mean-payoff value of 4. However, in order to do so, Adam needs to cooperate with Eve at all three corner vertices. Also note that if he does cooperate in at least one of these vertices then Eve can secure a payoff value of at least $\frac{11}{3}$.

Proof of Theorem 6.2 for LimInf *and mean-payoff.* For clarity, we focus on the

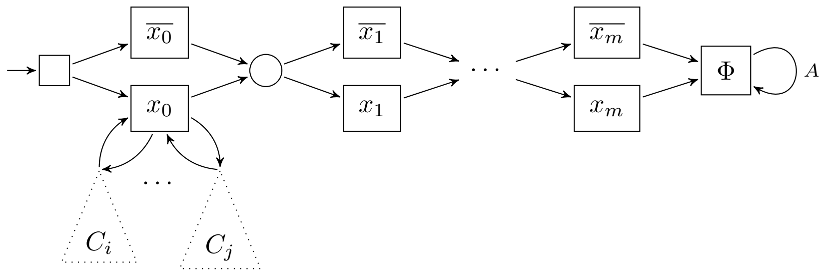

Figure 6.3: Depiction of the reduction from QBF.

non-strict regret threshold problem for the moment. We will later comment on how to adapt the construction for the strict case. Our reduction works for arbitrary values A, B, C, and r, satisfying the following constraints:

- $A < B < C$,
- $\frac{2C+B}{3} - A < r$ so that Eve wins if Φ is true,
- $C - A \geq r$ so that Adam wins if Φ is false, and
- $C - B < r$ so that he never helps Eve in the clause gadgets.

For concreteness, let us consider $A = 2$, $B = 3$, $C = 4$, and $r = 2$.

We first describe the value-choosing part of the game (see Figure 6.3). $V_\exists$ contains vertices for every existentially quantified variable from the QBF and $V \setminus V_\exists$ contains vertices for every universally quantified variable. At each of this vertices, there are two outgoing edges with weight 0 corresponding to a choice of truth value for the variable. For the variable x_i vertex, the **true** edge leads to a vertex from which Eve can choose to move to any of the clause gadgets corresponding to clauses where the literal x_i occurs (see dotted incoming edge in Figure 6.2) or to advance to x_{i+1}. The **false** edge construction is similar, while leading to the literal $\overline{x_i}$ rather than to x_i. From the vertices encoding the choice of truth value for x_n Eve can either visit the clause gadgets for it or move to a "final" vertex $\Phi \in V_\exists$. This final vertex has a self-loop with weight $A = 2$.

To conclude the proof, we describe the strategy of Eve which ensures the desired property if the QBF is satisfiable and a strategy of Adam which ensures the property is falsified otherwise.

Assume the QBF is true. It follows that there is a strategy of the existential player in the QBF game such that for any strategy of the universal player the QBF will be true after they both choose values for the variables. Eve now follows this strategy while visiting all clause gadgets corresponding to occurrences of chosen literals. At every gadget clause she visits she chooses to enter the gadget. If Adam now decides to take the weight $C = 4$ edge, Eve can achieve a mean-payoff value of $\frac{11}{3}$ or a LimInf value of $B = 3$ by staying in the gadget. In this case the claim trivially holds since the highest obtainable payoff in the constructed arena is 4 (for both functions). We therefore focus in the case where Adam chooses to take Eve back to the vertex from which she entered the gadget. She can now go to the next clause gadget and repeat. Thus, when the play reaches vertex Φ, Eve must have visited every clause gadget and Adam has chosen to disallow a weight 4 edge in every gadget. Now Eve can ensure a payoff value of 2 by going to Φ. As she has witnessed that in every clause gadget there is at least one vertex in which Adam is not helping her, alternative strategies might have ensured a mean-payoff of at most $\frac{11}{3}$ and a LimInf value of at most 3. Thus, her regret is less than $r = 2$.

Conversely, if the universal player had a winning strategy (or, in other words, the QBF was not true) then the strategy of Adam consists in following this strategy in choosing values for the variables and taking Eve out of clause gadgets if she ever enters one. If the play arrives at Φ we have that there is at least one clause gadget that was not visited by the play. We note there is an alternative strategy of Eve which, by choosing a different valuation of some variable, reaches this clause gadget and with the help of Adam achieves value 4. Hence, this strategy of Adam ensures regret of exactly 2. If Eve avoids reaching Φ then she can ensure a value of at most 0, which means an even greater regret for her.

Strict threshold. It is not hard to see that if we find a valuation for r, A, B, C which meets the first restriction and the last three having changed from strict to non-strict, and vice versa, we can get a reduction that works for the non-strict regret threshold problem. That is, find values such that

- $A < B < C$,
- $\frac{2C+B}{3} - A \leq r$ so that Eve wins if Φ is true,
- $C - A > r$ so that Adam wins if Φ is false, and
- $C - B \leq r$ so that he never helps Eve in the clause gadgets.

For example, one could consider $A = 5$, $B = 7$, $C = 10$ and $r = 4$. □

We will now show that the same holds for discounted-sum games. A new version of the clause-gadget is needed for this payoff function. The new gadget is depicted in Figure 6.4

Proof of Theorem 6.2 for discounted sum. Our reduction works for values of λ, r, A, B, and C such that the following constraints are met:

(i) $A < B < C$,

(ii) $\lambda^2 \left(\frac{C}{1-\lambda}\right) - \lambda^{2nm-2}\left(C + \lambda^2 \frac{B}{1-\lambda}\right) < r$,

(iii) $\lambda^{2nm-2}\left(C + \lambda^2 \frac{B}{1-\lambda}\right) > \lambda^2 \left(\frac{C\frac{1-\lambda^4}{1-\lambda}}{1-\lambda^8}\right)$,

(iv) $\lambda^2\left(C + \lambda^2 \frac{B}{1-\lambda}\right) - \lambda^{2nm}\left(\frac{A}{1-\lambda}\right) < r$, and

(v) $\lambda^{2nm-2}\left(\frac{C}{1-\lambda}\right) - \lambda^{2nm}\left(\frac{A}{1-\lambda}\right) \geq r$.

We remark that the constraints depend on the number m of variables in Φ and the number of clauses, n, as well. (See end of proof for a sample concrete assignment.) In the sequel we focus on the strict threshold problem, it will be clear how to adapt the construction in order to obtain the result for the non-strict version of the problem.

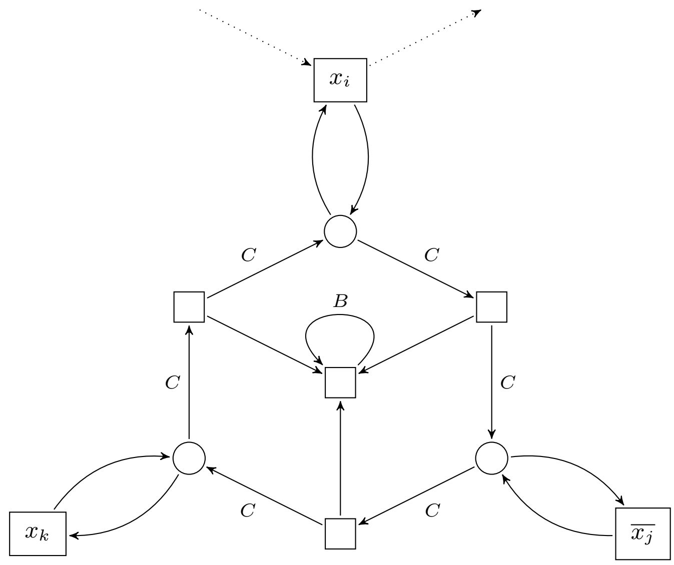

Figure 6.4: Clause gadget for the QBF reduction for clause $x_i \vee \neg x_j \vee x_k$.

The value-choosing part of the game is the same as the one for LimInf and mean-payoff games (see Figure 6.3).

We now describe the strategy of Eve which ensures the desired property if the QBF is true, and a strategy of Adam which ensures the property is falsified otherwise.

Assume the QBF is true. Eve follows the strategy of the existential player to win the QBF game. This allows her to visit all clause gadgets corresponding to occurrences of chosen literals. At every gadget clause she visits she chooses to enter the gadget. If Adam now decides to take the weight C edge, Eve can go to the center-most vertex and obtain a payoff of at least

$$\lambda^{2nm-2}\left(C + \lambda^2 \frac{B}{1-\lambda}\right),$$

with equality holding if Adam helps her at the very last clause visit of the very last variable gadget. In this case, the claim holds by (i). We therefore focus in the case where Adam chooses to take Eve back to the vertex from which she entered the gadget. She can now go to the next clause gadget and repeat. Thus, when the play reaches vertex Φ, Eve must have visited every clause gadget and Adam has chosen to disallow a weight C edge in every gadget. Now Eve can ensure a payoff value of

$$\lambda^{2nm}\left(\frac{A}{1-\lambda}\right)$$

by going to Φ. As she has witnessed that in every clause gadget there is at least one vertex in which Adam is not helping her, alternative strategies might have ensured a payoff of at most $\lambda^2(C + \lambda^2 \frac{B}{1-\lambda})$, by playing to the center of some clause gadget, or

$$\lambda^2 \left(\frac{C \frac{1-\lambda^4}{1-\lambda}}{1-\lambda^8} \right)$$

by playing in and out of some adjacent clause gadgets. By (*iii*), we know it suffices to show that the former is still not enough to make the regret of Eve at least r. Thus, from (*iv*), we get that her regret is less than r.

Conversely, if the universal player had a winning strategy (or, in other words, the QBF were false) then the strategy of Adam consists in following this strategy in choosing values for the variables and taking Eve out of clause gadgets if she ever enters one. If the play arrives at Φ we have that there is at least one clause gadget that was not visited by the play. We note there is an alternative strategy of Eve which, by choosing a different valuation of some variable, reaches this clause gadget and with the help of Adam achieves value of at least $\lambda^{2nm-2}(\frac{C}{1-\lambda})$. Hence, by (*v*), this strategy of Adam ensures regret of at least r. If Eve avoids reaching Φ then she can ensure a value of at most 0, which means an even greater regret for her.

Sample assignment. For completeness, we give one assignment of the positive rationals λ, r, A, B, and C which satisfies the inequalities. It will be obvious the chosen values can be encoded into a polynomial number of bits w.r.t. n and m.

We can assume, w.l.o.g., that $2 \leq 2m \leq n$. Intuitively, we want values such that (*i*) $A < B < C$ and such that the discount factor λ is close enough to 1 so that going to the center of a clause gadget at the end of the value-choosing rounds, is preferable for Adam compared to doing some strange path between adjacent clauses—this is captured by item (*iii*). A λ which is close to 1 also gives us item (*v*) from (*i*). In order to ensure Eve wins if she does visit the center of a clause gadget, we also would like to have $C - A < r\lambda^{-2}(1-\lambda)$, which would imply items (*ii*) and (*iv*) from the inequality list. It is not hard to see that the following assignment satisfies all the inequalities:

- $\lambda := 1 - \frac{1}{2^{n^3}}$,
- $A := 2$,
- $B := 3$,
- $C := 4$, and
- $r := 3(2^{n^6} - 1)$.

This concludes our proof. □

6.2 Upper Bounds for Prefix-Independent Functions

An interesting fact regarding the miminization of regret against a positional adversary is that, contrary to the case when Adam can use any strategy, Eve may benefit from using memory. As an example, let us consider a mean-payoff game played on the arena from Figure 6.5. Now, if Eve stays in u forever, she will obtain a payoff of 1. If, however, she first goes to v, at any point, she can

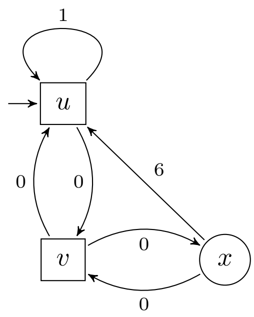

Figure 6.5: Example weighted arena.

still come back to u and obtain a payoff of 1. In this second case, she will have witnessed whether Adam plays from v to u or from v to x. Since Adam is only allowed to use positional strategies, he must commit to one of this two choices and repeat it every time v is visited. It follows that the best strategy for Eve, in terms of regret-minimization, is to first visit v to confirm Adam is not going to u—thus allowing the edge (x, u). If Adam is, indeed, not allowing that edge, she can always go back to u. But, if Adam goes to u then Eve can achieve a mean-payoff value of 2: the highest obtainable payoff in this arena. In both cases she has no regret. Additionally, it is easy to see that no positional strategy for Eve can achieve regret 0.

We provide a polynomial space algorithm to solve the regret threshold problem using, essentially, the same intuition from the example above. More specifically, for a given game we construct a new game in which Adam is forced to play positionally and such that the regret of the original game is a function of the antagonistic value of the new one. In order to force Adam to play positional strategies, we encode into the vertices the set of choices he has already made. Although the construction yields an exponential arena, we argue that this can be done on-the-fly, and only a polynomial number of vertices need to be considered.

Theorem 6.3. *For payoff functions* Inf, Sup, LimInf, LimSup, $\underline{\mathsf{MP}}$, *and* $\overline{\mathsf{MP}}$, *the regret of a game played against a positional adversary can be computed in polynomial space.* ■

Let $\mathcal{G} = (V, V_\exists, v_I, E, w)$ be a weighted arena. For a set of edges $D \subseteq E$, we denote by $\mathcal{G}\restriction D$ the weighted arena $(V, V_\exists, v_I, D, w)$. Also, for an edge $(s,t) \in E$ we define $E_\forall(st) := \{(u, v) \in E \mid \text{if } u = s \text{ then } v = t \text{ or } u \in V_\exists\}$. Intuitively, $E_\forall(e)$ is the subset of edges determine by the positional behavior of Adam we have already witnessed (by traversing e).

Given a weighted arena $\mathcal{G}$, we construct a new weighted arena $\hat{\mathcal{G}}$ such that we have that $-\mathbf{aVal}(\hat{\mathcal{G}})$ is equivalent to the regret of $\mathcal{G}$.

The vertices of $\hat{\mathcal{G}}$ encode the choices made by Adam. For a subset of edges $D \subseteq E$, let $\mathcal{G}\restriction D$ denote the weighted arena $(V, V_\exists, D, w, v_I)$. The new weighted arena $\hat{\mathcal{G}}$ is the tuple $(\hat{V}, \hat{V_\exists}, \hat{E}, \hat{w}, \hat{v_I})$ where

(i) $\hat{V} = V \times \mathcal{P}(E)$;

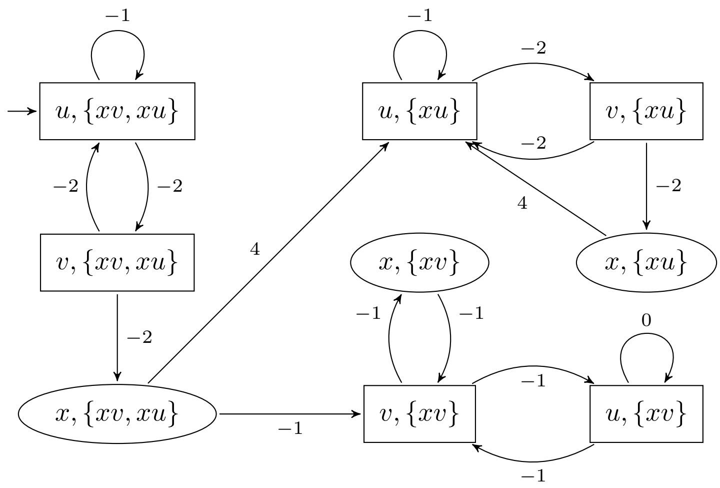

Figure 6.6: Weighted arena, constructed from Figure 6.5 w.r.t the $\underline{\mathsf{MP}}$ payoff function. In the edge set component only edges leaving Adam nodes are depicted.

(ii) $\hat{V_\exists} = \{(v, D) \in \hat{V} \mid v \in V_\exists\}$;

(iii) $\hat{v_I} = (v_I, E)$;

(iv) $\hat{E}$ contains the edge $\big((u, C), (v, D)\big)$ if and only if $(u, v) \in E$ and $D = C \cap E_\forall(uv)$;

(v) $\hat{w}\big((u, C), (v, D)\big) = w(u, v) - \mathbf{cVal}(\mathcal{G}{\restriction}D)$.

The application of this transformation for the weighted arena from Figure 6.5 w.r.t. to the $\underline{\mathsf{MP}}$ payoff function is given in Figure 6.6.

Consider a play $\hat{\pi} = (v_0, C_0)(v_1, C_1)\dots$ in $\hat{\mathcal{G}}$. We denote by $[\hat{\pi}]_{\mathbf{k}}$, for $k \in \{1, 2\}$, the sequence $\langle c_{k,i}\rangle_{i\geq 0}$, where $c_{k,i}$ is the k-th component of the i-th pair from $\hat{\pi}$. Observe that $[\hat{\pi}]_{\mathbf{1}}$ is a valid play in $\mathcal{G}$. Also observe that $E \supseteq C_j \supseteq C_{j+1}$ for all j. Hence $[\hat{\pi}]_{\mathbf{2}}$ is an infinite descending chain of finite subsets, and therefore $\lim\,[\hat{\pi}]_{\mathbf{2}}$ is well-defined. Finally, we define $\mathbf{c}(\hat{\pi}) := \mathbf{cVal}(\mathcal{G}{\restriction}\lim\,[\hat{\pi}]_{\mathbf{2}})$. The following result relates the value of a play in $\hat{\mathcal{G}}$ to the value of the corresponding play in $\mathcal{G}$.

Lemma 6.1. *For payoff functions* $\mathsf{LimInf}, \mathsf{LimSup}, \underline{\mathsf{MP}}, \overline{\mathsf{MP}}$ *and for any play* $\hat{\pi}$ *in* $\hat{\mathcal{G}}$ *we have that* $\mathbf{Val}(\hat{\pi}) = \mathbf{Val}([\hat{\pi}]_{\mathbf{1}}) - \mathbf{c}(\hat{\pi})$. ■

Proof. We first establish the following intermediate result. It follows from the existence of $\lim\,[\hat{\pi}]_{\mathbf{2}}$ and the definition of $\mathbf{c}(\cdot)$ that:

$$\limsup_{n\to\infty} \frac{1}{n}\sum_{i=0}^{n-1} \mathbf{cVal}(\mathcal{G}{\restriction}C_i) = \liminf_{n\to\infty} \frac{1}{n}\sum_{i=0}^{n-1} \mathbf{cVal}(\mathcal{G}{\restriction}C_i) = \mathbf{c}(\hat{\pi}). \tag{6.1}$$

We now show that the result holds for $\underline{\mathsf{MP}}$.

$$\begin{aligned}
\mathbf{Val}(\hat{\pi}) &= \liminf_{n\to\infty}\left(\frac{1}{n}\sum_{i=0}^{n-1}(w(v_i, v_{i+1}) - \mathbf{cVal}(\mathcal{G}\restriction C_j))\right) && \text{defs. of } \mathbf{Val}(\cdot), \hat{w}\\
&= \mathbf{Val}([\hat{\pi}]_{\mathbf{1}}) - \limsup_{n\to\infty}\frac{1}{n}\sum_{j=0}^{n-1}\mathbf{cVal}(\mathcal{G}\restriction C_j) && \text{def. of } \mathbf{Val}(\cdot)\\
&= \mathbf{Val}([\hat{\pi}]_{\mathbf{1}}) - \mathbf{c}(\hat{\pi}) && \text{from Eq. (6.1)}
\end{aligned}$$

The proofs for the other payoff functions are almost identical (for LimInf and LimSup replace the use of Equation (6.1) by Equation (6.2)).

$$\limsup_{i\to\infty}\mathbf{cVal}(\mathcal{G}\restriction C_i) = \liminf_{i\to\infty}\mathbf{cVal}(\mathcal{G}\restriction C_i) = \mathbf{c}(\hat{\pi}). \tag{6.2}$$

□

We now describe how to translate winning strategies for either player from $\hat{\mathcal{G}}$ back to $\mathcal{G}$, *i.e.* given an optimal maximizing (minimizing) strategy for Eve (Adam) in $\hat{\mathcal{G}}$ we construct the corresponding optimal regret minimizing strategy (memoryless regret maximizing counter-strategy) for Eve (Adam) in $\mathcal{G}$. For clarity, we follow this same naming convention throughout this section: again, we say a strategy is an optimal maximizing (minimizing) strategy when we speak about antagonistic and cooperative games, we say a strategy is an optimal regret maximizing (regret minimizing) when we speak about regret games. When this does not suffice, we explicitly state which kind of game we are speaking about.

Let $\hat{\varepsilon} \in \mathfrak{S}_\exists(\hat{\mathcal{G}})$ be an optimal maximizing strategy of Eve in $\hat{\mathcal{G}}$ and $\hat{\alpha} \in \mathfrak{S}_\forall(\hat{\mathcal{G}})$ be an optimal minimizing strategy of Adam. Indeed, in [EM79] it was shown that mean-payoff games are positionally determined. We will now define a strategy for Eve in $\mathcal{G}$ that, for every play prefix s, constructs a valid play prefix $\hat{s}$ in $\hat{\mathcal{G}}$ and plays as $\hat{\varepsilon}$ would in $\hat{\mathcal{G}}$ for $\hat{s}$. More formally, for a play prefix s from $\mathcal{G}$, denote by $[s]_{\mathbf{1}}^{-1}$ the corresponding sequence of vertex and edge-set pairs in $\hat{\mathcal{G}}$ (indeed, it is the inverse function of $[\cdot]_{\mathbf{1}}$, which is easily seen to be bijective). Define $\sigma \in \mathfrak{S}_\exists(\mathcal{G})$ as follows: $\sigma(s) = [\hat{\varepsilon}([s]_{\mathbf{1}}^{-1})]_{\mathbf{1}}$ for all play prefixes $s \in V^* \cdot V_\exists$ in $\mathcal{G}$ consistent with a positional strategy of Adam.

For a fixed strategy of Eve we can translate the optimal minimizing strategy of Adam in $\hat{\mathcal{G}}$ into an optimal memoryless regret maximizing counter-strategy of his in $\mathcal{G}$. Formally, for an arbitrary strategy σ for Eve in $\mathcal{G}$, define $\hat{\sigma} \in \mathfrak{S}_\exists(\hat{\mathcal{G}})$ as follows: $\hat{\sigma}(\hat{s}) = \sigma([\hat{s}]_{\mathbf{1}})$ for all $\hat{s} \in \hat{V}^* \cdot \hat{V}_\exists$. Let τ_σ be an optimal (positional) maximizing strategy for Adam in $\mathcal{G}\restriction \lim\,[\pi_{\hat{\sigma}\hat{\alpha}}]_{\mathbf{2}}$.

It is not hard to see the described strategy of Eve ensures a regret value of at most $-\mathbf{aVal}(\hat{\mathcal{G}})$. Slightly less obvious is the fact that for any strategy of Eve, the counter-strategy τ_σ of Adam is such that $\sup_{\sigma'\in\mathfrak{S}_\exists}\mathbf{Val}_{\mathcal{G}}(\sigma', \tau_\sigma) - \mathbf{Val}_{\mathcal{G}}(\sigma, \tau_\sigma) \geq -\mathbf{aVal}(\hat{\mathcal{G}})$.

Lemma 6.2. *For payoff functions* LimInf, LimSup, $\underline{\mathsf{MP}}$, *and* $\overline{\mathsf{MP}}$*:*

$$\mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{S}_\forall^1}(\mathcal{G}) = -\mathbf{aVal}(\hat{\mathcal{G}}).$$

■

Proof. The proof is decomposed into two parts. First, we describe a strategy $\sigma \in \mathfrak{S}_\exists(\mathcal{G})$ which ensures a regret value of at most $-\mathbf{aVal}(\hat{\mathcal{G}})$. Second, we show that for any $\sigma \in \mathfrak{S}_\exists(\mathcal{G})$ there is a $\tau \in \mathfrak{S}^1_\forall(\mathcal{G})$ such that

$$\sup_{\sigma' \in \mathfrak{S}_\exists} \mathbf{Val}_{\mathcal{G}}(\sigma', \tau) - \mathbf{Val}_{\mathcal{G}}(\sigma, \tau) \geq -\mathbf{aVal}(\hat{\mathcal{G}}).$$

The result follows.

We have already mentioned earlier that for a play $\hat{\pi}$ in $\hat{\mathcal{G}}$ we have that $[\hat{\pi}]_{\mathbf{1}}$ is a play in $\mathcal{G}$. Let $\mathrm{PPref}(\mathcal{G})$ denote the set of all play prefixes consistent with a positional strategy for Adam in $\mathcal{G}$. It is not difficult to see that $[\cdot]_{\mathbf{1}}$ is indeed a bijection between plays of $\hat{\mathcal{G}}$ and plays of $\mathcal{G}$ consistent with positional strategies for Adam.

It follows from the determinacy of antagonistic games defined by the payoff functions considered in this work that there are optimal strategies for Eve and Adam that ensure a payoff of, respectively, at least and at most a value $\mathbf{aVal}(\hat{\mathcal{G}})$ against any strategy of the opposing player. Let $\hat{\varepsilon} \in \mathfrak{S}_\exists(\hat{\mathcal{G}})$ be an optimal maximizing strategy of Eve in $\hat{\mathcal{G}}$ and $\hat{\alpha} \in \mathfrak{S}_\forall(\hat{\mathcal{G}})$ be an optimal minimizing strategy of Adam.

First Part. Define a strategy σ from $\mathfrak{S}_\exists(\mathcal{G})$ as follows: $\sigma(s) = [\hat{\varepsilon}([s]_{\mathbf{1}}^{-1})]_{\mathbf{1}}$ for all $s \in \mathrm{PPref}(\mathcal{G}) \cdot V_\exists$. We claim that

$$\mathbf{reg}^\sigma_{\mathfrak{S}_\exists, \mathfrak{S}^1_\forall}(\mathcal{G}) \leq -\mathbf{aVal}(\hat{\mathcal{G}}).$$

Towards a contradiction, assume there are $\tau \in \mathfrak{S}^1_\forall(\mathcal{G})$ and $\sigma' \in \mathfrak{S}_\exists(\mathcal{G})$ such that

$$\mathbf{Val}_{\mathcal{G}}(\sigma', \tau) - \mathbf{Val}_{\mathcal{G}}(\sigma, \tau) > -\mathbf{aVal}(\hat{\mathcal{G}}).$$

Define a strategy $\hat{\tau} \in \mathfrak{S}_\forall(\hat{\mathcal{G}})$ as follows: $\hat{\tau}(\hat{s}) = \tau([\hat{s}]_{\mathbf{1}})$ for all $\hat{s} \in \hat{V}^* \cdot \widehat{V \setminus V_\exists}$. From the definition of $\hat{\varepsilon}$ and our assumption we get that

$$\mathbf{Val}_{\mathcal{G}}(\sigma', \tau) - \mathbf{Val}_{\mathcal{G}}(\sigma, \tau) > -\mathbf{aVal}(\hat{\mathcal{G}}) \geq -\mathbf{Val}_{\hat{\mathcal{G}}}(\hat{\varepsilon}, \hat{\tau}). \tag{6.3}$$

It is straightforward to verify that $[\pi_{\sigma\tau}]_{\mathbf{1}}^{-1} = \pi_{\hat{\varepsilon}\hat{\tau}}$. Therefore, from Lemma 6.1, we have:

$$\mathbf{Val}(\pi_{\sigma'\tau}) > \mathbf{Val}(\pi_{\sigma\tau}) - \mathbf{Val}(\pi_{\hat{\varepsilon}\hat{\tau}}) = \mathbf{cVal}(\mathcal{G}\restriction \lim\, [\pi_{\hat{\varepsilon}\hat{\tau}}]_{\mathbf{2}}). \tag{6.4}$$

At this point we note that, since τ is positional, it holds that $\mathbf{Val}_{\mathcal{G}}(\sigma', \tau)$ is at most the highest payoff value attainable in $\mathcal{G}$ restricted to the edges allowed by τ. Formally, if $E_\tau = \{(u, v) \in E \mid u \in V \setminus V_\exists \implies v = \tau(u)\}$ then $\mathbf{Val}(\pi_{\sigma'\tau}) \leq \mathbf{cVal}(\mathcal{G}\restriction E_\tau)$. Also, by construction of $\hat{\tau}$ we get that $E_\tau \subseteq \lim\, [\pi_{\hat{\varepsilon}\hat{\tau}}]_{\mathbf{2}}$. It should be clear that this implies $\mathbf{cVal}(\mathcal{G}\restriction \lim\, [\pi_{\hat{\varepsilon}\hat{\tau}}]_{\mathbf{2}}) \geq \mathbf{cVal}(\mathcal{G}\restriction E_\tau)$. This contradicts Equation (6.4).

Second Part. For the second part of the proof we require the following result which relates positional strategies for Adam in $\mathcal{G}$ that agree on certain vertices to strategies in sub-graphs defined by plays in $\hat{\mathcal{G}}$.

Claim 8. *Let $\sigma \in \mathfrak{S}_\exists(\mathcal{G})$ and $\tau, \tau' \in \mathfrak{S}^1_\forall(\mathcal{G})$. Then $\pi_{\sigma\tau} = \pi_{\sigma\tau'}$ if and only if $\tau' \in \mathfrak{S}^1_\forall(\mathcal{G}\restriction \lim\, [[\pi_{\tau\sigma}]_{\mathbf{1}}^{-1}]_{\mathbf{2}})$* ■

Proof. *(only if)* Note that by construction of $\hat{\mathcal{G}}$ we have that once Adam chooses an edge $\big((u,C),(v,D)\big)$ from a vertex $(v,C) \in \hat{V} \setminus \hat{V_\exists}$ then on any subsequent visit to a vertex $(u,C') \in \hat{V} \setminus \hat{V_\exists}$ he has no other option but to go to (v,C'). That is, his choice is restricted to be consistent with the history of the play. For a play $\hat{\pi}$ in $\hat{\mathcal{G}}$, it is clear that the sequence $[\hat{\pi}]_{\mathbf{2}}$ is the decreasing sequence of sets of edges consistent (for Adam) with the history of the play in the same manner. In particular, for any $\tau' \in \mathfrak{S}^1_\forall(\mathcal{G})$ and any play π in $\mathcal{G}$ consistent with τ' we have that τ' is a valid strategy for Adam in $\mathcal{G}\restriction E'$ where $E' = \lim\,[[\pi]_{\mathbf{1}}^{-1}]_{\mathbf{2}}$. As $\pi_{\sigma\tau} = \pi_{\sigma\tau'}$ is a play consistent with τ', the result follows.

(if) Suppose $\pi_{\sigma\tau} \neq \pi_{\sigma\tau'}$, and let v be the last vertex in their common prefix. As σ is common to both plays, we have $v \in V \setminus V_\exists$, and $\tau(v) \neq \tau'(v)$. In particular, $(v,\tau'(v)) \notin \lim\,[[\pi_{\tau\sigma}]_{\mathbf{1}}^{-1}]_{\mathbf{2}}$ so $\tau' \notin \mathfrak{S}^1_\forall(\mathcal{G}\restriction \lim\,[[\pi_{\tau\sigma}]_{\mathbf{1}}^{-1}]_{\mathbf{2}})$. □

For an arbitrary strategy σ for Eve in $\mathcal{G}$, define $\hat{\sigma} \in \mathfrak{S}_\exists(\hat{\mathcal{G}})$ as follows: $\hat{\sigma}(\hat{s}\cdot(v,D)) = (\sigma([\hat{s}\cdot(v,D)]_{\mathbf{1}}^{-1}), D)$ for all $\hat{s}\cdot(v,D) \in \hat{V}^* \cdot \hat{V_\exists}$. Let τ_σ be an optimal (positional) maximizing strategy for Adam in $\mathcal{G}\restriction \lim\,[\pi_{\hat{\sigma}\hat{\alpha}}]_{\mathbf{2}}$. We claim that for all $\sigma \in \mathfrak{S}_\exists(\mathcal{G})$ we have that

$$\sup_{\sigma' \in \mathfrak{S}_\exists} \mathbf{Val}_{\mathcal{G}}(\sigma',\tau_\sigma) - \mathbf{Val}_{\mathcal{G}}(\sigma,\tau_\sigma) \geq -\mathbf{aVal}(\hat{\mathcal{G}}).$$

Towards a contradiction, assume that for some $\sigma \in \mathfrak{S}_\exists(\mathcal{G})$ it is the case that for all $\sigma' \in \mathfrak{S}_\exists(\mathcal{G})$ the left hand side of the above inequality is strictly smaller than the right hand side. By definition of $\hat{\alpha}$ we then get the following inequality.

$$\sup_{\sigma' \in \mathfrak{S}_\exists} \mathbf{Val}_{\mathcal{G}}(\sigma',\tau_\sigma) - \mathbf{Val}_{\mathcal{G}}(\sigma,\tau_\sigma) < -\mathbf{aVal}(\hat{\mathcal{G}}) \leq -\mathbf{Val}_{\hat{\mathcal{G}}}(\hat{\sigma},\hat{\alpha}) \tag{6.5}$$

Using the above Claim it is easy to show that $[\pi_{\sigma\tau_\sigma}]_{\mathbf{1}} = \pi_{\hat{\sigma}\hat{\alpha}}$. Hence, by Equation (6.5) and Lemma 6.1 we get that:

$$\sup_{\sigma' \in \mathfrak{S}_\exists} \mathbf{Val}(\pi_{\sigma'\tau_\sigma}) < \mathbf{cVal}(\mathcal{G}\restriction \lim\,[\pi_{\hat{\sigma}\hat{\alpha}}]_{\mathbf{2}}) \tag{6.6}$$

However, by choice of τ_σ, we know that there is a strategy $\sigma'' \in \mathfrak{S}_\exists(\mathcal{G})$ such that $\mathbf{Val}(\pi_{\sigma''\tau_\sigma}) = \mathbf{cVal}(\mathcal{G}\restriction \lim\,[\pi_{\hat{\sigma}\hat{\alpha}}]_{\mathbf{2}})$. This contradicts Equation (6.6) and completes the proof of the Theorem. □

If $\mathcal{G}$ was constructed from a Inf or Sup game $\mathcal{H}$, then one could easily transfer the described strategy of Eve, σ into a strategy for her in $\mathcal{H}$ which achieves the same regret. In order to have a symmetric result we still lack the ability to transfer a strategy of Adam from $\hat{\mathcal{G}}$ to the original game $\mathcal{H}$. Consider a modified construction in which we additionally keep track of the minimal (resp. maximal) weight seen so far by a play, just like described in Section 5.3. Denote the corresponding game by $\tilde{\mathcal{G}}$. The vertex set $\tilde{V}$ of $\tilde{\mathcal{G}}$ is thus a set of triples of the form (v,C,x) where x is the minimal (resp. maximal) weight the play has witnessed. We observe that in the proof of the above result the intuition behind why we can transfer a strategy of Adam from $\hat{\mathcal{G}}$ back to $\mathcal{G}$ as a memoryless strategy, **although the vertices in $\hat{\mathcal{G}}$ already encode additional information**, is that once we have fixed a strategy of Eve in $\mathcal{G}$, this gives us enough information about the prefix of the play before visiting any Adam vertex. In other words, we construct a strategy of Adam tailored to spoil a specific strategy of Eve, σ, in $\mathcal{G}$ using the information we gather from $[\cdot]_{\mathbf{1}}^{-1}$ and his optimal strategy in $\hat{\mathcal{G}}$. These properties still hold in $\tilde{\mathcal{G}}$. Thus, we get the following result.

Lemma 6.3. *For payoff functions* Inf, Sup*:* $\mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{S}^1_\forall}(\mathcal{G}) = -\mathbf{aVal}(\tilde{\mathcal{G}})$. ■

We recall a result from [EM79] which gives us an algorithm for computing the value $\mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{S}^1_\forall}(\mathcal{G})$ in polynomial space. In [EM79] the authors show that the value of a mean-payoff game $\mathcal{G}$ is equivalent to the value of a finite *cycle-forming game* $\Gamma_\mathcal{G}$ played on $\mathcal{G}$. The game is identical to the mean-payoff game except that it is finite. The game is stopped as soon as a cycle is formed and the value of the game is given by the mean-payoff value of the cycle.

Proposition 6.1 (Finite Mean-Payoff Game [EM79])**.** *The value of a mean-payoff game $\mathcal{G}$ is equal to the value of the finite cycle-forming game $\Gamma_\mathcal{G}$ played on the same weighted arena.* ■

Let us summarize the argument we have thus far presented.

Proof of Theorem 6.3. It follows from results in [AR14] that LimInf and LimSup games are equivalent to their finite cycle-forming game. Together with Proposition 6.1, this means one can use an alternating Turing machine to compute the value of a game and that said machine will stop in time bounded by the length of the largest simple cycle in the arena. We note the length of the longest simple path in $\hat{\mathcal{G}}$ is bounded by $|V|(|E|+1)$. Hence, we can compute the winner of $\hat{\mathcal{G}}$ in alternating polynomial time. Since APTIME = PSPACE, and the regret value of $\mathcal{G}$ is computable in constant time from $\hat{\mathcal{G}}$ (see Lemma 6.2) this concludes the proof of Theorem 6.3. □

Memory requirements for Eve. It follows from our algorithms for computing regret in this variant that Eve only requires strategies with exponential memory. Examples where exponential memory is necessary can be easily constructed.

Corollary 5. *For all payoff functions* Sup, Inf, LimSup, LimInf, $\underline{\mathsf{MP}}$, *and* $\overline{\mathsf{MP}}$, *for all game graphs $\mathcal{G}$, there exists m which is $\mathcal{O}(2^{|E|}|V|)$ such that:*

$$\mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{S}^1_\forall}(\mathcal{G}) = \mathbf{Reg}_{\mathfrak{S}^m_\exists,\mathfrak{S}^1_\forall}(\mathcal{G}).$$

■

6.3 Upper Bounds for Discounted Sum

In this section we consider the problem of computing the (minimal) regret when Adam is restricted to playing positional strategies.

Theorem 6.4. *Deciding if the regret value is less than a given threshold (strictly or non-strictly), playing against positional strategies of Adam, is in* EXPSPACE*; in* PSPACE*, if λ is not part of the input or if $r = 0$.* ■

Proof. The result follows from Lemma 6.5 and Lemma 6.12. □

Playing against an Adam, when he is restricted to playing memoryless strategies gives Eve the opportunity to learn some of Adam's strategic choices. However, due to its decaying nature, with the discounted-sum payoff function Eve must find a balance between exploring too quickly, thereby presenting lightly discounted alternatives; and learning too slowly, thereby heavily discounting her eventual payoff.

A similar approach to the one we have adopted in Section 5.4 can be used to obtain an algorithm for this setting. The claimed lower bound follows from Theorem 6.1.

6.3.1 Deciding 0-regret

As in the previous section, we will reduce the problem of deciding if the game has regret value 0 to that of determining the winner of a safety game. It will be obvious that if no regret-free strategy for Eve exists in the original game, then we can construct, for any strategy of hers, a positional strategy of Adam which ensures non-zero regret. Hence, we will also obtain a lower bound on the regret of the game in the case Adam wins the safety game.

We extend $E_\forall(\cdot)$ to play prefixes $\varrho = v_0 \dots v_n$ by (recursively) defining $E_\forall(\varrho) := E_\forall(\varrho[..n-1]) \cap E_\forall(v_{n-1}v_n)$. If π is a play, then $E \supseteq E_\forall(\pi[..i]) \supseteq E_\forall(\pi[..j])$ for all $0 \leq i \leq j$. Hence, since E is finite, the value $E_\forall(\pi) := \lim_{i \geq 0} E_\forall(\pi[..i])$ is well-defined. Remark that $E_\forall(\pi)$ does not restrict edges leaving vertices of Eve. The following properties directly follow from our definitions.

Lemma 6.4. *Let π be a play or play prefix consistent with a positional strategy for Adam. It then holds that:*

(i) for every $v \in V \setminus V_\exists$ there is some edge $(v, \cdot) \in E_\forall(\pi)$,

(ii) π is consistent with $\tau \in \mathfrak{S}^1_\forall(\mathcal{G})$ if and only if $\tau \in \mathfrak{S}^1_\forall(\mathcal{G} \restriction E_\forall(\pi))$, and

(iii) every strategy $\tau \in \mathfrak{S}^1_\forall(\mathcal{G} \restriction E_\forall(\pi))$ is also an element from $\mathfrak{S}^1_\forall(\mathcal{G})$.

■

To be able to decide whether regret-free strategies for Eve exist, we define a new safety game. The arena we consider is $\hat{\mathcal{G}} := (\hat{V}, \hat{V_\exists}, \hat{v_I}, \hat{E})$ where $\hat{V} := V \times \mathcal{P}(E)$, $\hat{V_\exists} := V_\exists \times \mathcal{P}(E)$, $\hat{v_I} := (v_I, E)$, and $\hat{E}$ contains the edge $((u, C), (v, D))$ if and only if $(u, v) \in E$ and $D = C \cap E_\forall(uv)$.

Lemma 6.5. *Deciding if the regret value is* 0*, playing against positional strategies of Adam, is in* PSPACE. ■

Proof. A safety game is constructed as in the proof of Lemma 5.1. Here, we consider $\tilde{\mathcal{G}}$ and the set of bad edges $\tilde{\mathcal{B}} := \{((u, C), (v, D)) \in \hat{E} \mid u \in V_\exists$ and $\exists \tau \in \mathfrak{S}^1_\forall(\mathcal{G} \restriction C), w(u, v) + \lambda \mathbf{cVal}^v(\mathcal{G} \times \tau) < \mathbf{cVal}^u_{\neg v}(\mathcal{G} \times \tau)\}$. We then have the safety game $\tilde{\mathcal{G}} = (\hat{V}, \hat{V_\exists}, \hat{v_I}, \hat{E}, \tilde{\mathcal{B}})$. Note that there is an obvious bijective mapping from plays (and play prefixes) in $\tilde{\mathcal{G}}$ to plays (prefixes) in $\mathcal{G}$ which are consistent with a positional strategy for Adam. One can then show the following properties hold:

Claim 9. *If $\tau \in \mathfrak{S}_\forall(\tilde{\mathcal{G}})$ is a winning strategy for Adam in $\tilde{\mathcal{G}}$, then for all $\sigma \in \mathfrak{S}_\exists(\mathcal{G})$, there exist $t_{\tau\sigma} \in \mathfrak{S}^1_\forall(\mathcal{G})$ and $s_{\tau\sigma} \in \mathfrak{S}_\exists(\mathcal{G})$ such that* $\mathbf{Val}(s_{\tau\sigma}, t_{\tau\sigma}) - \mathbf{Val}(\sigma, t_{\tau\sigma}) \geq \lambda^{|V|(|E|+1)}$
$\min\{\mathbf{cVal}^u_{\neg v}(\mathcal{G} \times \tau) - w(u, v) - \lambda \mathbf{cVal}^v(\mathcal{G} \times \tau) \mid ((u, C), (v, D)) \in \tilde{\mathcal{B}}, \tau \in \mathfrak{S}^1_\forall(\mathcal{G} \restriction C)\}$.
■

The claim follows from positional determinacy of safety games together with Lemma 6.4.

Claim 10. *If $\sigma \in \mathfrak{S}_\exists(\tilde{\mathcal{G}})$ is a winning strategy for Eve in $\tilde{\mathcal{G}}$, then there is $s_\sigma \in \mathfrak{S}_\exists(\mathcal{G})$ such that $\mathbf{reg}^{s_\sigma}_{\mathfrak{S}_\exists,\mathfrak{S}^1_\forall}(\mathcal{G}) = 0$.* ■

It then follows from the determinacy of safety games that Eve wins the safety game $\tilde{\mathcal{G}}$ if and only if she has a regret-free strategy.

We observe that simple cycles in $\tilde{\mathcal{G}}$ have length at most $|V|(|E|+1)$. Thus, we can simulate the safety game until we complete a cycle and check that all traversed edges are good, all in alternating polynomial time. Indeed, an alternating Turing machine can simulate the cycle and then (universally) check that for all edges, for all positional strategies of the Adam, the inequality holds. □

It remains to convince the reader Claim 9 and Claim 10 hold.

Proof of Claim 9. We will now argue that if $\tau \in \mathfrak{S}_\forall(\tilde{\mathcal{G}})$ is a winning strategy for Adam in $\tilde{\mathcal{G}}$, then for all $\sigma \in \mathfrak{S}_\exists(\mathcal{G})$, there exist $t_{\tau\sigma} \in \mathfrak{S}^1_\forall(\mathcal{G})$ and $s_{\tau\sigma} \in \mathfrak{S}_\exists(\mathcal{G})$ such that $\mathbf{Val}(s_{\tau\sigma}, t_{\tau\sigma}) - \mathbf{Val}(\sigma, t_{\tau\sigma})$ is at least

$$\lambda^{|V|(|E|+1)} \min_{\substack{((u,C),(v,D))\in\tilde{\mathcal{B}}\\ \tau\in\mathfrak{S}^1_\forall(\mathcal{G}\restriction C)}} \{\mathbf{cVal}^u_{\neg v}(\mathcal{G}\times\tau) - w(u,v) - \lambda\mathbf{cVal}^v(\mathcal{G}\times\tau)\}. \quad (6.7)$$

The argument is straightforward and based on the bijection between plays from $\mathcal{G}$, which are consistent with positional strategies of Adam, and plays in $\tilde{\mathcal{G}}$. Recall that safety games are positionally determined. That is, either Eve has a positional strategy which allows her to perpetually avoid the unsafe edges against any strategy for Adam, or Adam has a positional strategy which ensures that—regardless of the behavior of Eve—the play eventually traverses some unsafe edge. Thus, since we assume $\tau \in \mathfrak{S}_\forall(\tilde{\mathcal{G}})$ is winning for Adam in $\tilde{\mathcal{G}}$ we can assume that τ is in fact a positional strategy for Adam in $\tilde{\mathcal{G}}$. Now consider an arbitrary strategy σ for Eve in $\mathcal{G}$. We note, once more, that τ is a strategy for Adam in $\mathcal{G}$, not only in $\tilde{\mathcal{G}}$. Furthermore, τ is a positional strategy for Adam in $\mathcal{G}$. Conversely, σ is a valid strategy for Eve in $\tilde{\mathcal{G}}$. These facts follow from the definition of $E_\forall(\cdot)$ and construction $\tilde{\mathcal{G}}$. Since τ is winning for Adam in $\tilde{\mathcal{G}}$, the play $\pi_{\sigma\tau}^{\sim}$ traverses an unsafe edge. In fact, since τ is positional, the unsafe edge is necessarily traversed in at most $|V|(|E|+1)$ steps—that is, at most the length of the longest simple path in $\tilde{\mathcal{G}}$. Let us write $(\tilde{v}_i, \tilde{v}_{i+1}) = ((v_i, C_i), (v_{i+1}, C_{i+1}))$ for the traversed unsafe edge at step $i \leq |V|(|E|+1)$. By definition of $\tilde{\mathcal{B}}$ we have that there exists $t_{\tau\sigma} \in \mathfrak{S}^1_\forall(\mathcal{G}\restriction C_i)$ such that

$$\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G}\times t_{\tau\sigma}) - w(v_i, v_{i+1}) - \lambda\mathbf{cVal}^{v_i}(\mathcal{G}\times t_{\tau\sigma}).$$

We now move from $\tilde{\mathcal{G}}$ back to the original game $\mathcal{G}$. Henceforth, we consider the play $\pi_{\sigma\tau} = v_0 v_1 \ldots$ in $\mathcal{G}$ which corresponds to $\tilde{\pi}_{\sigma\tau} = (v_0, C_0)(v_1, C_1)\ldots$ in $\tilde{\mathcal{G}}$. It is easy to see that $\pi_{\sigma\tau}[..i]$ is consistent with $t_{\tau\sigma}$. Hence, $\pi_{\sigma t_{\tau\sigma}}$ traverses edge (v_i, v_{i+1}) corresponding to bad edge $(\tilde{v}_i, \tilde{v}_{i+1})$ in $\tilde{\mathcal{G}}$. Finally, by determinacy of discounted-sum games and by virtue of $\mathcal{G}\times t_{\tau\sigma}$ being a finite weighted arena, we have that there is a strategy $s_{\tau\sigma} \in \mathfrak{S}_\exists(\mathcal{G}\times t_{\tau\sigma})$ such that $\mathbf{Val}^{v_i}_{\mathcal{G}}(s_{\tau\sigma}, t_{\tau\sigma}) = \mathbf{cVal}^{v_i}(\mathcal{G}\times t_{\tau\sigma})$. It then follows from the definition of $\mathbf{cVal}$ and $\mathcal{G}\times s_{\tau\sigma}$ that $\mathbf{Val}^{v_I}_{\mathcal{G}}(s_{\tau\sigma}, t_{\tau\sigma}) - \mathbf{Val}^{v_I}_{\mathcal{G}}(\sigma, t_{\tau\sigma})$ is at least the value from Equation (6.7), just as required. □

Proof of Claim 10. Let us show that if $\sigma \in \mathfrak{S}_\exists(\tilde{\mathcal{G}})$ is a winning strategy for Eve in $\tilde{\mathcal{G}}$, then there is $s_\sigma \in \mathfrak{S}_\exists(\mathcal{G})$ such that $\mathbf{reg}^{s_\sigma}_{\mathfrak{S}_\exists,\mathfrak{S}^1_\forall}(\mathcal{G}) = 0$. The intuition behind the argument is the same as for the proof of Claim 5. However, in this case we first need to describe how to construct the strategy for Eve in $\mathcal{G}$ from a strategy for her in $\tilde{\mathcal{G}}$.

A regret-free strategy from $\tilde{\mathcal{G}}$. Observe that, by construction of $\tilde{\mathcal{G}}$, for any vertex $(u, C) \in \hat{V}_\exists$ and any edge $(u, v) \in E$ there is exactly one corresponding edge in $\tilde{\mathcal{G}}$: $((u, C), (v, C))$. Given a vertex (u, C) from $\tilde{\mathcal{G}}$, denote by $[(u, C)]_\mathbf{1}$ the vertex u. Now, given a strategy $\sigma \in \mathfrak{S}_\exists(\tilde{\mathcal{G}})$ we define $s_\sigma \in \mathfrak{S}_\exists(\mathcal{G})$ as follows

$$s_\sigma(v_0v_1v_2\dots) = [\sigma((v_0, C_0)(v_1, C_1 = C_0 \cap E_\forall(v_0v_1))(v_2, C_1 \cap E_\forall(v_1v_2))\dots)]_\mathbf{1}$$

where $C_0 = E$. It follows from the fact that we have a bijective mapping from plays in $\tilde{\mathcal{G}}$ to plays in $\mathcal{G}$ which are consistent with positional strategies for Adam, that s_σ is a valid strategy for Eve in $\mathcal{G}$ when playing against a positional adversary. Additionally, it is easy to see that s_σ can be realized using finite memory only. The memory required corresponds to the subsets of E. The current memory element is determined by the applying the operator $E_\forall(\cdot)$ to the current play prefix.

Now that we have our strategy s_σ for Eve in $\mathcal{G}$, we proceed by proving the analogue of Claim 6 in this setting.

Claim 11. *If $\sigma \in \mathfrak{S}_\exists(\tilde{\mathcal{G}})$ is a winning strategy for Eve in $\tilde{\mathcal{G}}$, then*

$$\forall \tau \in \mathfrak{S}^1_\forall(\mathcal{G}), \forall i \geq 0 : \mathbf{Val}(\pi_{s_\sigma\tau}[i..] = v_i \dots) \geq \mathbf{cVal}^{v_i}(\mathcal{G} \times \tau). \tag{6.8}$$

■

Proof. To convince the reader that s_σ has the property from Equation (6.8), we consider the synchronized product of $\mathcal{G}$ and s_σ—that is, the synchronized product of $\mathcal{G}$ and the finite Mealy machine realizing s_σ. As s_σ is a finite memory strategy, then this product, which we denote in the sequel by $\mathcal{G} \times s_\sigma$, is finite. Now, towards a contradiction, suppose that Equation (6.8) does not hold for s_σ. That is, there is some $\tau \in \mathfrak{S}^1_\forall(\mathcal{G})$ for which the property fails. Further, let us consider an alternative (memoryless) strategy σ' of Eve which ensures $\mathbf{cVal}^v(\mathcal{G} \times \tau)$ from all $v \in V$. The latter exists by definition of $\mathbf{cVal}(\mathcal{G} \times \tau)$ and memoryless determinacy of discounted-sum games (see, *e.g.* [ZP96]).

Let $\mathcal{H}$ denote a copy of $\mathcal{G} \times s_\sigma$ where all edges induced by E from $\mathcal{G}$ are added—not just the ones allowed by s_σ—and $\mathcal{H}\restriction\sigma'$ denote the sub-graph of $\mathcal{H}$ where only edges allowed by σ' are left. Intuitively, both $\mathcal{G} \times s_\sigma$ and $\mathcal{H}\restriction\sigma'$ are sub-structures of $\tilde{\mathcal{G}}$ with a weight function $\tilde{w}$ lifted from w to the blown-up vertex set $\tilde{V}$. This is due to the way in which we constructed s_σ.

Since, by assumption, s_σ does not have the property of Equation (6.8) then the edges present in at least one vertex from $\mathcal{H}\restriction\sigma'$ and $\mathcal{G} \times \sigma$ differ. Note that such a vertex (u, C) is necessarily such that $u \in V_\exists$—and C is a "memory element" from the machine realizing s_σ corresponding to a subset of E obtained via $E_\forall(\cdot)$. Furthermore, from our definition of a strategy, we know that there is a single outgoing edge from it in both structures. Let us write (u, v)—instead of $((u, C), (v, D))$—for the edge in $\mathcal{G} \times s_\sigma$ and (u, v') for the edge in $\mathcal{H}\restriction\sigma'$. Recall that s_σ is winning for Eve in $\tilde{\mathcal{G}}$. Thus, we have that $(u, v) \notin \tilde{\mathcal{B}} =$

$\{((u,C),(v,D)) \in \hat{E} \mid u \in V_\exists$ and $\exists \tau' \in \mathfrak{S}^1_\forall(\mathcal{G}\restriction C), w(u,v) + \lambda \mathbf{cVal}^v(\mathcal{G} \times \tau') < \mathbf{cVal}^u_{\neg v}(\mathcal{G} \times \tau')\}$. It follows that

$$w(u,v) + \lambda \mathbf{cVal}^v(\mathcal{H} \times \tau) \geq \mathbf{cVal}^{v'}(\mathcal{H} \times \tau).$$

Thus, the strategy σ'' of Eve which takes (u,v) instead of (u,v') and follows σ' otherwise—indeed, this might mean σ'' is no longer memoryless—also achieves at least $\mathbf{cVal}^u(\mathcal{H} \times \tau)$ from u onwards. Notice that this process can be repeated for all vertices in which the two structures differ. Further, since both are finite, it will eventually terminate and yield a strategy of Eve which plays exactly as s_σ and for which, since τ was chosen arbitrarily, Equation (6.8) holds. Contradiction. □

It follows immediately that $\mathbf{reg}^{s_\sigma}_{\mathfrak{S}_\exists,\mathfrak{S}^1_\forall}(\mathcal{G}) = 0$. Indeed, if we suppose that this is not the case, then there exists a strategy $\sigma' \in \mathfrak{S}_\exists(\mathcal{G})$ such that

$$\exists \tau \in \mathfrak{S}^1_\forall(\mathcal{G}) : \mathbf{Val}(s_\sigma, \tau) < \mathbf{Val}(\sigma', \tau).$$

The above directly contradicts Claim 11. □

Corollary 6 (Corollary of Lemma 6.5)**.** *If no regret-free strategy for Eve exists in $\mathcal{G}$, then $\mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{S}^1_\forall}(\mathcal{G}) \geq b_\mathcal{G}$ where $b_\mathcal{G} := \lambda^{|V|(|E|+1)} \min\{\mathbf{cVal}^u_{\neg v}(\mathcal{G} \times \tau) - w(u,v) - \lambda \mathbf{cVal}^v(\mathcal{G} \times \tau) \mid ((u,C),(v,D)) \in \tilde{\mathcal{B}}$ and $\tau \in \mathfrak{S}^1_\forall(\mathcal{G}\restriction C)\}$.* ■

6.3.2 Deciding r-regret

In this section we present sufficient modifications to our definitions from Section 5.4 in order for the techniques used therein to be adapted for this case. Particularly, our notion of regret of a play and the safety game used to decide the existence of regret-free strategies need to take into account the fact that witnessing edges taken by Adam affects previously observed local regrets. That is, we formalize the intuition that alternative plays must also be consistent with the behavior of Adam that we have witnessed in the current play.

We are now ready to define the regret of a play in a game against a positional adversary. Given a play $\pi = v_0 v_1 \ldots$, we let

$$\mathbf{reg}(\pi) := \sup\{\lambda^i(\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G}\restriction E_\forall(\pi)) - \mathbf{Val}(\pi[i..]) \mid v_i \in V_\exists\} \cup \{0\}.$$

Consider now a play prefix $\varrho = v_0 \ldots v_j$. We let the regret of ϱ be

$$\max\{\lambda^i(\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G}\restriction E_\forall(\varrho[i..j])) - \mathbf{Val}(\varrho[i..j]) \mid 0 \leq i < j \text{ and } v_i \in V_\exists\} \cup \{0\}.$$

We will now re-prove Lemma 5.2 in the current setting.

Lemma 6.6. *For any strategy σ of Eve,*

$$\mathbf{reg}^\sigma_{\mathfrak{S}_\exists,\mathfrak{S}^1_\forall}(\mathcal{G}) = \sup\{\mathbf{reg}(\pi) \mid \pi \text{ is consistent with } \sigma \text{ and some } \tau \in \mathfrak{S}^1_\forall\}.$$

■

Proof. Consider any $\sigma, \sigma' \in \mathfrak{S}_\exists$ and $\tau \in \mathfrak{S}^1_\forall$ such that $\pi_{\sigma\tau} \neq \pi_{\sigma'\tau}$. Let us write $\pi_{\sigma\tau} = v_0 v_1 \dots$ and $\pi_{\sigma'\tau} = v'_0 v'_1 \dots$ and denote by ℓ the length of the longest common prefix of $\pi_{\sigma\tau}$ and $\pi_{\sigma'\tau}$. We claim that

$$\mathbf{cVal}^{v_\ell}_{\neg v_{\ell+1}}(\mathcal{G}{\restriction}E_\forall(\pi_{\sigma\tau})) - \mathbf{Val}(\pi_{\sigma\tau})[\ell..] \geq \mathbf{Val}(\pi_{\sigma'\tau}[\ell..]) - \mathbf{Val}(\pi_{\sigma\tau}[\ell..]). \tag{6.9}$$

Indeed, if we assume it is not the case, we then get that

$$\mathbf{cVal}^{v'_{\ell+1}}(\mathcal{G}{\restriction}E_\forall(\pi_{\sigma\tau})) < \mathbf{Val}(\pi_{\sigma'\tau}[\ell+1..]).$$

However, recall that $\mathcal{G} \times \tau$ is a sub-arena of $\mathcal{G}{\restriction}E_\forall(\pi_{\sigma\tau})$. Thus, the co-operative value Eve can obtain in the former, say by playing σ', must be at most that which she can obtain in the latter. Contradiction.

Note that there is another positional strategy τ' for Adam and a second alternative strategy σ'' for Eve which do give us equality for Equation (6.9). For this purpose, we choose τ' so that $\tau' \in \mathfrak{S}^1_\forall(\mathcal{G}{\restriction}E_\forall(\pi_{\sigma\tau}))$—so that $\pi_{\sigma\tau}$ is also consistent with τ', thus $E_\forall(\pi_{\sigma\tau}) = E_\forall(\pi_{\sigma\tau'})$ (see Lemma 6.4)—and also such that

$$\mathbf{cVal}^{v'_{\ell+1}}(\mathcal{G} \times \tau') = \mathbf{cVal}^{v'_{\ell+1}}(\mathcal{G}{\restriction}E_\forall(\pi_{\sigma\tau})).$$

We choose σ'' so that it follows σ for ℓ turns, goes to v', and then plays co-operatively with τ' from v'. More formally, let σ'' be a strategy for Eve such that $\pi_{\sigma\tau}[..\ell] = \pi_{\sigma''\tau}[..\ell]$ and therefore, by choice of τ', such that $\pi_{\sigma\tau'}[..\ell] = \pi_{\sigma''\tau'}[..\ell]$ and so that

$$\mathbf{Val}(\pi_{\sigma''\tau'}[\ell..]) = \mathbf{cVal}^{v'_{\ell+1}}(\mathcal{G} \times \tau').$$

It follows from Equation (6.9) and the above arguments that for all $\sigma \in \mathfrak{S}_\exists$, if there are $\tau \in \mathfrak{S}^1_\forall$ and $\sigma' \in \mathfrak{S}_\exists$ such that $\pi_{\sigma\tau} \neq \pi_{\sigma'\tau}$ then

$$\sup_{\tau,\sigma' : \pi_{\sigma\tau} \neq \pi_{\sigma'\tau}} \mathbf{Val}(\pi_{\sigma'\tau}[\ell..]) - \mathbf{Val}(\pi_{\sigma\tau}[\ell..]) = \mathbf{cVal}^{v_\ell}_{\neg v_{\ell+1}}(\mathcal{G}{\restriction}E_\forall(\pi_{\sigma\tau})) - \mathbf{Val}(\pi_{\sigma\tau}). \tag{6.10}$$

We are now able to prove the result. That is, for any strategy σ for Eve:

$$\begin{aligned}
&\sup\{\mathbf{reg}(\pi) \mid \pi \text{ is consistent with } \sigma \text{ and some } \tau \in \mathfrak{S}^1_\forall\} \\
&= \sup_{\tau \in \mathfrak{S}^1_\forall} \mathbf{reg}(\pi_{\sigma\tau} = v_0 v_1 \dots) && \text{def. of } \pi_{\sigma\tau} \\
&= \sup_{\tau \in \mathfrak{S}^1_\forall} \max \left\{ 0, \sup_{\substack{i \geq 0 \\ v_i \in V_\exists}} \lambda^i X_i \right\} && \text{def. of } \mathbf{reg}(\pi_{\sigma\tau}) \\
&\quad \text{where } X_i := \left(\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G}{\restriction}E_\forall(\pi_{\sigma\tau})) - \mathbf{Val}(\pi_{\sigma\tau}[i..]) \right) \\
&= \sup_{\tau \in \mathfrak{S}^1_\forall} \max \left\{ 0, \sup_{\sigma' : \pi_{\sigma\tau} \neq \pi_{\sigma'\tau}} \lambda^\ell Y_\ell \right\} && \text{by Eq. (6.10)} \\
&\quad \text{where } Y_\ell := (\mathbf{Val}(\pi_{\sigma'\tau}[\ell..]) - \mathbf{Val}(\pi_{\sigma\tau}[\ell..])) \\
&= \sup_{\tau \in \mathfrak{S}^1_\forall} \max \left\{ 0, \sup_{\sigma' : \pi_{\sigma\tau} \neq \pi_{\sigma'\tau}} (\mathbf{Val}(\sigma', \tau) - \mathbf{Val}(\sigma, \tau)) \right\} && \text{def. of } \mathbf{Val}(\cdot), \ell \\
&= \sup_{\tau \in \mathfrak{S}^1_\forall} \sup_{\sigma' \in \mathfrak{S}_\exists} (\mathbf{Val}(\sigma', \tau) - \mathbf{Val}(\sigma, \tau)) && 0 \text{ when } \pi_{\sigma\tau} = \pi_{\sigma'\tau}
\end{aligned}$$

as required. □

We will now state and prove a restricted version of Lemma 5.3. Intuitively, for a play π, we will not be able to consider a deviation with respect to a prefix of π. Rather, we are forced to take the co-operative value with respect to the set $E_\forall(\pi)$—that is, the edges consistent with any positional strategy Adam might be playing—even after the bound on where the best deviation occurs.

Lemma 6.7. *Let π be a play in $\mathcal{G}$ and suppose $0 < r \leq \mathbf{reg}(\pi)$. Let*

$$N(r) := \lfloor (\log r + \log(1-\lambda) - \log(2w_{\max}))/\log\lambda \rfloor + 1.$$

Then $\mathbf{reg}(\pi)$ is equal to

$$\max_{\substack{0 \leq i < N(r)\\ v_i \in V_\exists}} \{\lambda^i(\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G}\restriction E_\forall(\pi)) - \mathbf{Val}(\pi[i..N(r)])\} - \lambda^{N(r)}\mathbf{Val}(\pi[N(r)..]).$$

■

Proof. Observe that $N(r)$ is such that $\frac{2w_{\max}\lambda^{N(r)}}{1-\lambda} < r$. Hence, we have that for all $i \geq N(r)$ such that $v_i \in V_\exists$ it holds that $\lambda^i(\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G}) - \mathbf{Val}(\pi[i..])) \leq \frac{2w_{\max}\lambda^{N(r)}}{1-\lambda} < r$. Clearly, since $\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{H}) \leq \mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G})$ holds for any sub-arena $\mathcal{H}$ of $\mathcal{G}$, we have that

$$\lambda^i(\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G}\restriction E_\forall(\pi)) - \mathbf{Val}(\pi[i..])) \leq \frac{2w_{\max}\lambda^{N(r)}}{1-\lambda} < r.$$

It thus follows that

$$\begin{aligned}\mathbf{reg}(\pi) &= \sup\{\lambda^i(\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G}\restriction E_\forall(\pi)) - \mathbf{Val}(\pi[i..])) \mid i \geq 0 \text{ and } v_i \in V_\exists\}\\ &= \max_{\substack{0 \leq i < N(r)\\ v_i \in V_\exists}} \lambda^i \left(\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G}\restriction E_\forall(\pi))\mathbf{Val}(\pi[i..N(r)])\right)\\ &\quad - \lambda^{N(r)}\mathbf{Val}(\pi[N(r)..])\end{aligned}$$

as required. □

The main difference between the problem at hand and the one we solved in Section 5.4 is that, when playing against a positional adversary, information revealed to Eve in the present can affect the best alternatives to her current behavior. Some definitions are in order. Let $\varrho = v_0 \dots v_j$ be a play prefix. The *maximal-regret points* of ϱ, denoted by $\mathbf{MRP}(\varrho)$, is the set

$$\{0 \leq i < j \mid v_i \in V_\exists \text{ and } \lambda^i \left(\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G}\restriction E_\forall(\varrho[..j])) - \mathbf{Val}(\varrho[i..j])\right) = \mathbf{reg}(\varrho)\};$$

and the *maximal-regret strategies* of ϱ, written $\mathbf{MRS}(\varrho)$, is equal to

$$\left\{\tau \in \mathfrak{S}^1_\forall(\mathcal{G}\restriction E_\forall(\varrho[..j])) \mid \tau \text{ satisfies } \varphi\right\}$$

where

$$\varphi := \bigvee_{i \in \mathbf{MRP}(\varrho)} \mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G}\restriction E_\forall(\varrho[..j])) = \mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G} \times \tau).$$

The above definitions are meant to capture the intuition that, upon witnessing a new choice of Adam, we can reduce the size of the set of possible positional

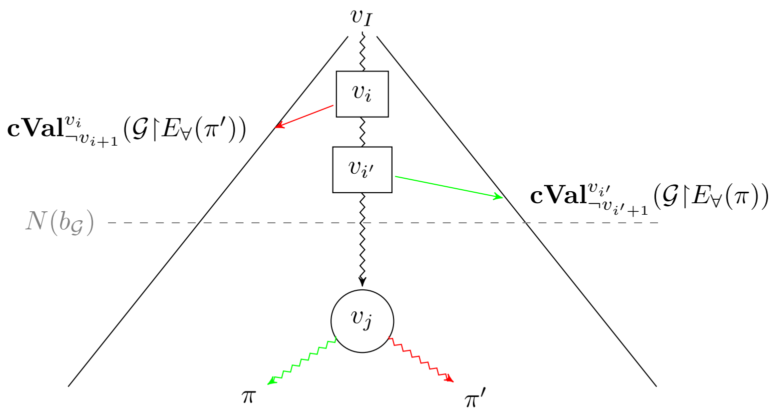

Figure 6.7: Let ϱ denote the play prefix $v_0 \dots v_j$. The alternative play from $v_{i'}$ is better than the one from v_i w.r.t ϱ. However, for play π' extending ϱ, the alternative play from v_i becomes better than the one from $v_{i'}$ if $\lambda^{i'-i}\mathbf{cVal}^{v_{i'}}_{\neg v_{i'+1}}(\mathcal{G}\restriction E_\forall(\pi'))$ is smaller than $\mathbf{cVal}^{v_i}_{\neg v_i+1}(\mathcal{G}\restriction E_\forall(\pi')) - \mathbf{Val}(\varrho[i..i'])$.

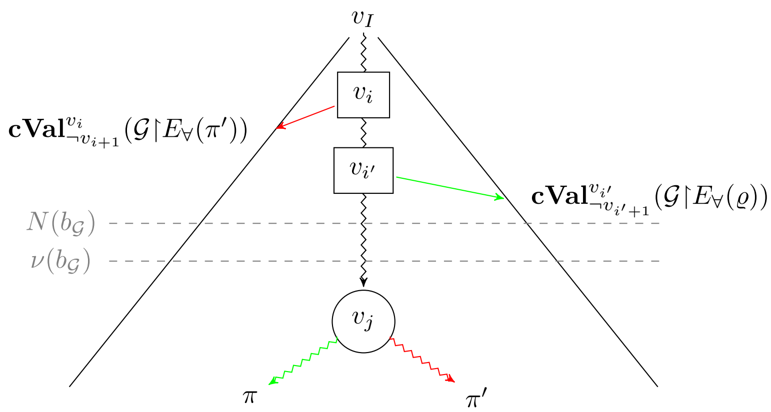

Figure 6.8: A play π' extending ϱ in a way such that $\mathfrak{S}^1_\forall(\mathcal{G}\restriction E_\forall(\pi')) \cap \mathbf{MRS}(\varrho) = \varnothing$ cannot have more regret than a play π extending ϱ for which $\mathfrak{S}^1_\forall(\mathcal{G}\restriction E_\forall(\pi)) \cap \mathbf{MRS}(\varrho) \neq \varnothing$—for ϱ longer than $\nu(b_\mathcal{G})$.

strategies he could be using. Consider a play prefix ϱ. The maximal-regret points of ϱ correspond to the positions at which best alternatives to ϱ occur. The maximal-regret strategies of ϱ is the set of positional strategies of Adam, ϱ consistent with them, such that at least one of the best alternatives to ϱ is consistent with them. Recall from Lemma 6.4 (*ii*) that a play prefix ϱ is consistent with a positional strategy $\tau \in \mathfrak{S}^1_\forall(\mathcal{G})$ if and only if $\tau \in \mathfrak{S}^1_\forall(\mathcal{G} \restriction E_\forall(\varrho))$. We can, therefore, think of the set of edges $E_\forall(\varrho)$ as representing the set of all positional strategies for Adam in $\mathcal{G}$ that ϱ is consistent with, *i.e.* $\{\tau \in \mathfrak{S}^1_\forall(\mathcal{G}) \mid \varrho \text{ is consistent with } \tau\}$. Let us write $\mathfrak{S}^1_\forall(\mathcal{G}, \varrho)$ for the set we just described. Let β be the value of one of the best alternatives to ϱ. If $\beta' < \beta$ is the value of one of the best alternatives to ϱ', then we know the best alternatives to ϱ are not consistent with any strategy from $\mathfrak{S}^1_\forall(\mathcal{G}, \varrho')$. Then, according to our definition of maximal-regret strategies, this also means that $\mathbf{MRS}(\varrho) \cap \mathfrak{S}^1_\forall(\mathcal{G}, \varrho') = \varnothing$. The converse is also true.

As an example, consider the situation depicted in Figure 6.7. If, from v_j, the play π' is obtained and we have that $\mathfrak{S}^1_\forall(\mathcal{G} \restriction E_\forall(\pi') \cap \mathbf{MRS}(\varrho)$ is empty, then the deviation from $v_{i'}$ might no longer be a best alternative. Indeed, there is no positional strategy of Adam which allows the deviation from $v_{i'}$ to obtain the value we assumed (from just looking at the prefix ϱ) and which is also consistent with π'. In order to deal with this, we need some more definitions.

Assume that $\mathbf{Reg}_{\mathfrak{S}_\exists, \mathfrak{S}^1_\forall}(\mathcal{G}) \geq b_\mathcal{G}$. For a play prefix $\varrho = v_0 \dots v_n$ with $n \geq N(b_\mathcal{G})$, let us define the value δ_ϱ (δ for *drop*) as

$$\min_{\substack{0 \leq i \leq j < N(b_\mathcal{G}) \\ \tau, \tau' \in \mathfrak{S}^1_\forall(\mathcal{G} \restriction E_\forall(\varrho))}} \left| \lambda^i \left(\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G} \times \tau) - \mathbf{Val}(\varrho[i..j]) \right) - \lambda^j \mathbf{cVal}^{v_j}_{\neg v_{j+1}}(\mathcal{G} \times \tau')) \right|.$$

Intuitively δ_ϱ is the minimal drop of the regret achievable by a better alternative (given the information we can extract from ϱ).

The smallest possible drop. Let us derive a universal lower bound on δ_ϱ for all ϱ of length at least $N(b_\mathcal{G})$. In order to do so we will recall "the shape" of the co-operative value of $\mathcal{G}$. Recall the **cVal** in a discounted-sum game can be obtained by supposing Eve controls all vertices and computing **aVal** instead. It then follows from positional determinacy of discounted-sum games that the **cVal** is achieved by a lasso in the arena $\mathcal{G}$. More formally, we know that there is a play π in $\mathcal{G}$ of the form

$$\pi = v_0 \dots v_{k-1}(v_k \dots v_\ell)^\omega$$

where $0 \leq k < \ell \leq |V|$, and such that $\mathbf{Val}(\pi) = \mathbf{cVal}^{v_0}(\mathcal{G})$. Let us write $\lambda = \frac{\alpha}{\beta}$ with $\alpha, \beta \in \mathbb{Z}$. One can then verify that

Lemma 6.8. *For all sub-arenas $\mathcal{H}$ of $\mathcal{G}$, for all vertices $v \in V$, there exists $N \in \mathbb{Z}$ such that $\mathbf{cVal}^v(\mathcal{H}) = \frac{N}{D}$ where $D := \beta^{|V|}(\beta^{|V|} - \alpha^{|V|})$.* ■

It then follows from the definition of δ_ϱ that:

Lemma 6.9. *For all play prefixes $\varrho = v_0 \dots v_n$ such that $n \geq N(b_\mathcal{G})$ we have that*

$$\delta_\varrho > \frac{1}{\beta^{N(b_\mathcal{G})} D}.$$

■

Formalizing our claims. We can now prove a replacement for Lemma 5.3 holds in this context.

Lemma 6.10. *Let π be a play in $\mathcal{G}$ and assume $\mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{S}^1_\forall}(\mathcal{G}) > 0$. Let $\nu(b_\mathcal{G})$ denote the value*

$$N(b_\mathcal{G}) + \left\lfloor \frac{\log(1-\lambda) - \log w_{\max} - (N(b_\mathcal{G}) + |V|)\log\beta - \log(\beta^{|V|} - \alpha^{|V|})}{\log\lambda} \right\rfloor + 1.$$

Then for all $\sigma \in \mathfrak{S}_\exists$,

$$\sup_{\tau\in\mathfrak{S}^1_\forall} \mathbf{reg}(\pi_{\sigma\tau}) = \sup_{\tau\in\mathfrak{S}^1_\forall} \mathbf{reg}(\pi_{\sigma\tau}[..\nu(b_\mathcal{G})]) - \lambda^{\nu(b_\mathcal{G})}\mathbf{Val}(\pi_{\sigma\tau}[\nu(b_\mathcal{G})..]).$$

■

Proof. Let us consider throughout this argument an arbitrary $\sigma \in \mathfrak{S}_\exists$. From Lemma 6.7 and the fact that $\nu(b_\mathcal{G})$ is such that $N(b_\mathcal{G})$, it follows that

$$\begin{aligned}
&\sup_{\tau\in\mathfrak{S}^1_\forall} \mathbf{reg}(\pi_{\sigma\tau} = v_0 v_1 \dots) \\
&= \sup_{\tau\in\mathfrak{S}^1_\forall} \max_{\substack{0\le i<\nu(b_\mathcal{G}) \\ v_i\in V_\exists}} \{\lambda^i(\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G}\restriction E_\forall(\pi_{\sigma\tau})) - \mathbf{Val}(\pi_{\sigma\tau}[i..\nu(b_\mathcal{G})])\} \\
&\quad - \lambda^{\nu(b_\mathcal{G})}\mathbf{Val}(\pi_{\sigma\tau}[\nu(b_\mathcal{G})..]).
\end{aligned}$$

Now, also note that $\nu(b_\mathcal{G})$ was chosen so that

$$\frac{w_{\max}\lambda^{\nu(b_\mathcal{G})}}{1-\lambda} < \frac{1}{\beta^{N(b_\mathcal{G})+|V|}D}.$$

Hence, for all $\tau' \in \mathfrak{S}^1_\forall$ if we write $\pi_{\sigma\tau'} = v'_0 \dots$, then for all $j \ge \nu(b_\mathcal{G})$ such that $v'_j \in V_\exists$ it holds that

$$-\frac{1}{\beta^{N(b_\mathcal{G})+|V|}D} < \lambda^i\mathbf{Val}(\pi_{\sigma\tau'}[i..])) < \frac{1}{\beta^{N(b_\mathcal{G})+|V|}D}.$$

It then follows from Lemma 6.9 and the definition of $\delta_{\pi_{\sigma\tau'}[..\nu(b_\mathcal{G})]}$ that, if there exists $\ell \ge \nu(b_\mathcal{G})$ such that for all $0 \le k \le \nu(b_\mathcal{G})$ with $v'_k \in V_\exists$

$$\mathbf{cVal}^{v'_k}_{\neg v'_{k+1}}(\mathcal{G}\restriction E_\forall(\pi[..\ell])) < \mathbf{cVal}^{v'_k}_{\neg v'_{k+1}}(\mathcal{G}\restriction E_\forall(\pi[..\nu(b_\mathcal{G})]))$$

then $\mathbf{reg}(\pi_{\sigma\tau'}) < \mathbf{reg}(\pi_{\sigma\tau''})$ for all $\tau'' \in \mathbf{MRS}(\pi'[..\nu(b_\mathcal{G})])$. This is due to the fact that that $\pi_{\sigma\tau''}[..\nu(b_\mathcal{G})] = \pi_{\sigma\tau'}[..\nu(b_\mathcal{G})]$ and

$$\mathbf{cVal}^{v'_k}_{\neg v'_{k+1}}(\mathcal{G}\times\tau'') = \mathbf{cVal}^{v'_k}_{\neg v'_{k+1}}(\mathcal{G}\restriction E_\forall(\pi_{\sigma\tau''}[..\nu(b_\mathcal{G})])).$$

The above implies that for all $\sigma \in \mathfrak{S}_\forall$ the value $\sup_{\tau\in\mathfrak{S}^1_\forall}\mathbf{reg}(\pi_{\sigma\tau} = v_0\dots)$ equals

$$\begin{aligned}
\max\{&\mathbf{cVal}^{v_i}_{\neg v_{i+1}}(\mathcal{G}\restriction E_\forall(\pi_{\sigma\tau}[..\nu(b_\mathcal{G})])) - \lambda^{\nu(b_\mathcal{G})}\mathbf{Val}(\pi_{\sigma\tau}[..\nu(b_\mathcal{G})]) \\
&: 0 \le i \le N(b_\mathcal{G}) \text{ and } v_i \in V_\exists\}
\end{aligned}$$

and therefore (by definition of regret of a prefix) we have that

$$\sup_{\tau\in\mathfrak{S}^1_\forall} \mathbf{reg}(\pi_{\sigma\tau}) = \sup_{\tau\in\mathfrak{S}^1_\forall} \mathbf{reg}(\pi_{\sigma\tau}[..\nu(b_\mathcal{G})]) - \lambda^{\nu(b_\mathcal{G})}\mathbf{Val}(\pi_{\sigma\tau}[\nu(b_\mathcal{G})..]).$$

as required. □

Putting everything together. Let us go back to our example to illustrate how to use $\nu(b_{\mathcal{G}})$ and the drop of a prefix. Consider now the situation from Figure 6.8. Recall we have assumed π' is a play extending ϱ with $\mathfrak{S}^1_\forall(\mathcal{G}\restriction E_\forall(\pi')) \cap \mathbf{MRS}(\varrho) = \varnothing$. It follows that all best alternatives to π' achieve a payoff strictly smaller than $\mathbf{cVal}^{v'}_{\neg v_{i'+1}}(\mathcal{G}\restriction E_\forall(\varrho))$. Thus, the regret of π' can only be bigger than the regret of a play π with $\mathfrak{S}^1_\forall(\mathcal{G}\restriction E_\forall(\pi)) \cap \mathbf{MRS}(\varrho) \neq \varnothing$ if the minimal index $k > j$ such that $\mathfrak{S}^1_\forall(\mathcal{G}\restriction E_\forall(\pi'[..j])) \cap \mathbf{MRS}(\varrho) = \varnothing$—*i.e.* the turn at which Adam revealed he was not playing a strategy from $\mathbf{MRS}(\varrho)$—is small enough. In other words, the drop in the value of the best alternative has to be compensated by a similar drop in the value obtained by Eve, and the discount factor makes this impossible after some number of turns.

Lemma 6.11. *If $\mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{S}^1_\forall}(\mathcal{G}) \geq b_{\mathcal{G}}$ then $\mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{S}^1_\forall}(\mathcal{G})$ is equal to*

$$\inf_{\sigma\in\mathfrak{S}_\exists} \sup\{\mathbf{reg}(\pi[..\nu(b_{\mathcal{G}})]) - \lambda^{\nu(b_{\mathcal{G}})}\mathbf{aVal}^{\hat{u}}(\hat{\mathcal{H}})$$
$$: \pi = v_0 v_1 \ldots \text{ cons. with } \sigma \text{ and some } \tau \in \mathfrak{S}^1_\forall\}$$

where

- $\hat{u} := (v_{\nu(b_{\mathcal{G}})}, E_\forall(\pi[..\nu(b_{\mathcal{G}})]))$ *and*
- $\hat{\mathcal{H}} := \hat{\mathcal{G}}\restriction\{((C,u),(D,v)) \mid \mathfrak{S}^1_\forall(\mathcal{G}\restriction D) \cap \mathbf{MRS}(\pi[..\nu(b_{\mathcal{G}})]) \neq \varnothing\}$.

■

Proof. First, note that if $\mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{S}^1_\forall}(\mathcal{G}) > 0$ then there cannot be any regret-free strategies for Eve in $\mathcal{G}$ when playing against a positional adversary. It then follows from Corollary 6 that $\mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{S}^1_\forall}(\mathcal{G}) \geq b_{\mathcal{G}}$.

Now using Lemma 6.10 together with the definition of the regret of a play we get that $\mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{S}^1_\forall}(\mathcal{G})$ is equal to

$$\inf_{\sigma\in\mathfrak{S}_\exists} \sup\{\mathbf{reg}(\pi[..\nu(b_{\mathcal{G}})]) - \lambda^{\nu(b_{\mathcal{G}})}\mathbf{Val}(\pi[\nu(b_{\mathcal{G}})..]) \mid \pi \text{ cons. } \sigma \text{ and some } \tau \in \mathfrak{S}^1_\forall\}.$$

Finally, note that it is in the interest of Eve to maximize $\lambda^{\nu(b_{\mathcal{G}})}\mathbf{Val}(\pi[\nu(b_{\mathcal{G}})..])$ in order to minimize regret. Conversely, Adam tries to minimize the same value with a strategy from $\mathbf{MRS}(\pi[..\nu(b_{\mathcal{G}})])$: critically, the strategy is such that the prefix $\pi[..\nu(b_{\mathcal{G}})]$ is consistent with it. Thus, we can replace it by the antagonistic value from $\pi[\nu(b_{\mathcal{G}})..]$ discounted accordingly. In this setting we also want to force Adam to play a positional strategy which is consistent with deviations before $N(b_{\mathcal{G}})$ which achieve the assumed regret of the prefix $\pi[..\nu(b_{\mathcal{G}})]$. More formally, we have

$$\begin{aligned}
&\inf_{\sigma\in\mathfrak{S}_\exists} \sup_{\tau\in\mathfrak{S}^1_\forall} \mathbf{reg}(\pi_{\sigma\tau}[..\nu(b_{\mathcal{G}})]) - \lambda^{\nu(b_{\mathcal{G}})}\mathbf{Val}(\pi_{\sigma\tau}[\nu(b_{\mathcal{G}})..]) \\
&= \inf_{\substack{\sigma\in\mathfrak{S}_\exists \\ \sigma'\in\mathfrak{S}_\exists}} \sup_{\substack{\tau\in\mathfrak{S}^1_\forall \\ \tau'\in\mathbf{MRS}(\pi_{\sigma\tau}[..\nu(b_{\mathcal{G}})])}} \mathbf{reg}(\pi_{\sigma\tau}[..\nu(b_{\mathcal{G}})]) - \lambda^{\nu(b_{\mathcal{G}})}\mathbf{Val}(\sigma',\tau') \\
&= \inf_{\sigma\in\mathfrak{S}_\exists} \sup_{\tau\in\mathfrak{S}_\forall} \mathbf{reg}(\pi_{\sigma\tau}[..\nu(b_{\mathcal{G}})]) \\
&\quad + \inf_{\sigma'\in\mathfrak{S}_\exists} \sup_{\tau'\in\mathbf{MRS}(\pi_{\sigma\tau}[..\nu(b_{\mathcal{G}})])} \left(-\lambda^{\nu(b_{\mathcal{G}})}\mathbf{Val}(\sigma',\tau')\right).
\end{aligned}$$

It should be clear that the RHS term of the sum is equivalent to

$$-\lambda^{\nu(b_{\mathcal{G}})}\mathbf{aVal}^{\hat{u}}(\hat{\mathcal{H}})$$

as required. □

The above result allows us to claim a PSPACE algorithm (EXPSPACEwhen λ is given as part of the input) to compute the regret of a game. As in Section 5.4, we simulate the game using an alternating machine which halts in at most a pseudo-polynomial number of steps which depends on $\nu(b_{\mathcal{G}})$ and, in turn, on $b_{\mathcal{G}}$. After that, we must compute the antagonistic value of $\hat{\mathcal{G}}$. As a first step, however, we compute the safety game $\tilde{\mathcal{G}}$ and determine its winner.

Lemma 6.12. *Computing the regret value of a game, playing against a positional adversary, can be done in time $\mathcal{O}(\max\{|V|(|E|+1), \nu(b_{\mathcal{G}})\})$ with an alternating Turing machine.* ■

The memory requirements for Eve are as follows:

Corollary 7. *Let $\eta := |\Delta|^d$ where $d = \max\{|V|(|E|+1), \nu(b_{\mathcal{G}})\}$. It then holds that $\mathbf{Reg}_{\mathfrak{S}_\exists^\eta, \mathfrak{S}_\forall^1}(\mathcal{G}) = \mathbf{Reg}_{\mathfrak{S}_\exists, \mathfrak{S}_\forall^1}(\mathcal{G})$.* ■

Chapter 7

Minimizing Regret Against Eloquent Adversaries

In this chapter we study the regret threshold problem when Eve plays against an *eloquent adversary*—so called because a strategy of the adversary corresponds to an infinite word.

Let us illustrate the usefulness of the variant in which Adam is restricted to play word strategies. Assume that we need to design a system embedded into an environment that produces disturbances: if the sequence of disturbances produced by the environment is independent of the behavior of the system, then it is natural to model this sequence not as a function of the state of the system but as a temporal sequence of events, *i.e.* a *word* on the alphabet of the disturbances. Clearly, if the sequences are not the result of an antagonistic process, then minimizing the regret against all disturbance sequences is an adequate solution concept to obtain a reasonable system and may be preferable to a system obtained from a strategy that is optimal under the antagonistic hypothesis. This disturbance-handling embedded system example was first given in [DF11]. In that work, the authors introduce *remorsefree strategies.* Such strategies correspond to strategies which minimize regret in games with ω-regular objectives. They do not establish lower bounds on the complexity of realizability or synthesis of remorsefree strategies and they focus on word strategies of Adam only.

In [HP06], Henzinger and Piterman introduce the notion of *good-for-games automata.* A non-deterministic automaton is good for solving games if it fairly simulates the equivalent deterministic automaton. We show below that our notion of regret minimization for word strategies extends this notion to the quantitative setting (Proposition 7.2). Our definitions give rise to a natural notion of approximate determinization for weighted automata on infinite words.

In [AKL10], Aminof et al. introduce the notion of *approximate determinization by pruning* for weighted sum automata over finite words. For $\alpha \in (0, 1]$, a weighted sum automaton is *α-determinizable by pruning* if there exists a finite state strategy to resolve non-determinism and that constructs a run whose value is at least α times the value of the maximal run of the given word. So, they consider a notion of approximation which is a *ratio.* We will show that our concept of regret, when Adam plays word strategies only, defines instead a notion of approximation with respect to the *difference* metric for weighted automata (Proposition 7.1). There are other differences with their work. First, we consider infinite words while they consider finite words. Second, we study a general notion of regret minimization problem in which Eve can use any strategy while they restrict their study to fixed memory strategies only and leave the problem open when the memory is not fixed a priori.

Finally, the main difference between these related works and our work is that we study the Inf, Sup, LimInf, LimSup, $\underline{\mathsf{MP}}$, $\overline{\mathsf{MP}}$, and DS_λ measures while they consider the total sum measure or Boolean objectives.

Contributions. In this chapter we continue our study of the regret threshold problem when Adam is restricted to playing word strategies. For payoff functions which admit deterministic automata, there is a simple algorithm to compute a strategy for Eve to minimize her regret. We now briefly sketch it. For any given quantitative automaton, we first determinize it—that is, we translate it into a deterministic automaton which realizes the same mapping from infinite words to reals—and then play a *quantitative simulation game* on the product of both automata. The latter game consists in Adam playing letters and moving a token on the deterministic automaton (note that he does not have a real choice, as the machine is deterministic) and Eve responding by resolving non-determinism in the original automaton. Eve wins the game if, the infinite run in the original non-deterministic automaton has value r-away from the value assigned by the deterministic one to the word given by Adam. Otherwise, Adam wins and Eve loses. Observe that the determinization step is indeed crucial. Hence, for the mean-payoff and discounted-sum functions an alternative solution is needed. Indeed, mean-payoff and discounted-sum are not determinizable in general.

The structure of this chapter differs somewhat from the previous two. We will first give additional definitions required for us to make the transition from games played on graphs to games played on automata. Then, in Section 7.1 we will give lower bounds for the regret threshold problem when considering a threshold of 0 and in the general case as well. The strongest of these lower bounds is the one obtained for mean payoff, where we show the problem to be undecidable even if further restricting Eve to play using only finite memory. Next, in Section 7.2 we show that the threshold 0 case coincides with determining whether Eve has a regret-free strategy and that both are in NP. Finally, we describe algorithms for the general threshold problem, first for prefix-independent functions in Section 7.3, and then for discounted sum in Section 7.4.

Additional definitions

We say that a strategy of Adam is a *word strategy* if his strategy can be expressed as a function

$$\tau : \mathbb{N} \to \{1, \ldots, \max_{v \in V} \mathsf{deg}^+(v)\}.$$

Intuitively, we consider an order on the successors of each Adam vertex. On every turn, the strategy τ of Adam will tell him to move to the $\tau(i)$-th successor (or to a sink state, if its outdegree is less than $\tau(i)$) of the vertex according to the fixed order. We denote by $\mathfrak{W}_\forall$ the set of all such strategies for Adam. A game in which Adam plays word strategies can be reformulated as a game played on a weighted automaton $\mathcal{G} = (Q, q_I, A, \Delta, w)$ and strategies of Adam—of the form $\tau : \mathbb{N} \to A$—determine a sequence of input symbols to which Eve has to react by choosing Δ-successor states starting from q_I. In this setting a strategy of Eve which minimizes regret defines a run by resolving the non-determinism of Δ in $\mathcal{G}$, and ensures the difference of value given by the constructed run is minimal w.r.t. the value of the best run on the word spelled out by Adam. (Observe that Eve and Adam here take the opposite roles as compared to usual games played on automata. That is, Eve here resolves non-determinism while Adam "spells", instead of the other way around.)

For instance, consider once more the example given in Chapter 4. If all vertices in Figure 4.1 are replaced by states, Eve can choose the successor of v_1 regardless of what letter Adam plays and, from v_2 and v_3, Adam chooses the successor by choosing to play a or b. Furthermore, his choice of letter tells Eve what would have happened had the play been at the other state.

Relation to other concepts. Let us extend the definitions of *approximation, embodiment* and *refinement* from [AKL10] to the setting of ω-words. Consider two weighted automata $\mathcal{A} = (Q_\mathcal{A}, q_I, A, \Delta_\mathcal{A}, w_\mathcal{A})$ and $\mathcal{B} = (Q_\mathcal{B}, q_I, A, \Delta_\mathcal{B}, w_\mathcal{B})$ and let $d : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a *metric*.[1] We say $\mathcal{B}$ *(strictly)* α*-approximates* $\mathcal{A}$ *(with respect to d)* if $d(\mathcal{B}(w), \mathcal{A}(w)) \leq \alpha$ (resp. $d(\mathcal{B}(w), \mathcal{A}(w)) < \alpha$) for all words $w \in A^\omega$. We say $\mathcal{B}$ *embodies* $\mathcal{A}$ if $Q_\mathcal{A} \subseteq Q_\mathcal{B}$, $\Delta_\mathcal{A} \subseteq \Delta_\mathcal{B}$ and $w_\mathcal{A}$ agrees with $w_\mathcal{B}$ on $\Delta_\mathcal{A}$. For an automaton $\mathcal{A} = (Q, q_I, A, \Delta, w)$ and an integer $k \geq 0$, the k-refinement of $\mathcal{A}$ is the automaton obtained by refining the state space of $\mathcal{A}$ using k Boolean variables. Intuitively, this corresponds to having 2^k copies of every state, with each copy of p transitioning to all copies of q with a if $(p, a, q) \in \Delta$. The automaton $\mathcal{A}$ is said to be *(strictly)* (α, k)*-determinizable by pruning* (DBP, for short) if the k-refinement of $\mathcal{A}$ embodies a deterministic automaton which (strictly) α-approximates $\mathcal{A}$. The next result follows directly from the above definitions.

Proposition 7.1. *For $\alpha \in \mathbb{Q}$, $k \in \mathbb{N}$, a weighted automaton $\mathcal{G}$ is (strictly) (α, k)-DBP (w.r.t. the difference metric) if and only if $\mathbf{Reg}_{\mathfrak{S}_\exists^{2^k}, \mathfrak{W}_\forall}(\mathcal{G}) \leq \alpha$ (resp. $\mathbf{Reg}_{\mathfrak{S}_\exists^{2^k}, \mathfrak{W}_\forall}(\mathcal{G}) < \alpha$).* ■

In [HP06] the authors define *good-for-games automata*. Their definition is based on a game which is played on an ω-automaton by Spoiler and Simulator. We propose the following generalization of the notion of good for games automata for weighted automata. A weighted automaton $\mathcal{A}$ is *(strictly)* α*-good for games* if Simulator, against any word $x \in A^\omega$ spelled by Spoiler, can resolve non-determinism in $\mathcal{A}$ so that the resulting run has value v and $d(v, \mathcal{A}(x)) \leq \alpha$ (resp. $d(v, \mathcal{A}(x)) < \alpha$), for some metric d. We summarize the relationship that follows from the definition in the following result:

Proposition 7.2. *For $\alpha \in \mathbb{Q}$, a weighted automaton $\mathcal{G}$ is (strictly) α-good for games (w.r.t. the difference metric) if and only if $\mathbf{Reg}_{\mathfrak{S}_\exists, \mathfrak{W}_\forall}(\mathcal{G}) \leq \alpha$ (resp. $\mathbf{Reg}_{\mathfrak{S}_\exists, \mathfrak{W}_\forall}(\mathcal{G}) < \alpha$).* ■

[1] The metric used in [AKL10] is the ratio measure.

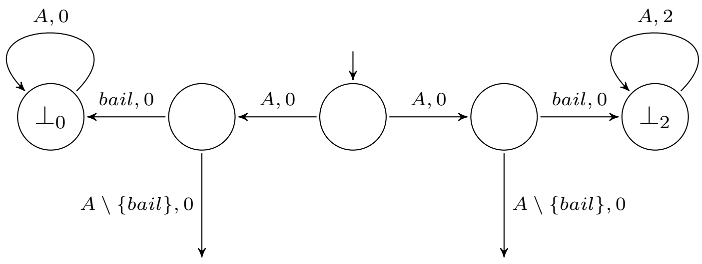

Figure 7.1: Initial gadget used in most reductions.

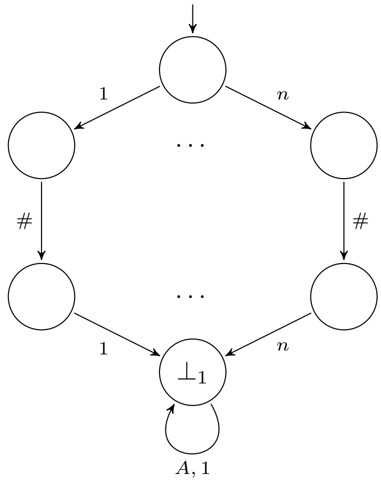

Figure 7.2: Clause-choosing gadget for the SAT reduction. There are as many paths from top to bottom ($\bot_1$) as there are clauses (n).

7.1 Lower Bounds

We will progress gradually by first showing how the 0-regret threshold problem is NP-hard for all payoff functions, and working our way towards the undecidability of the general threshold problem for mean payoff.

The following result is shown to be true using an adaptation of the NP-hardness proof from [AKL10].

Theorem 7.1. *Consider a fixed regret threshold $r \in \mathbb{Q}$ and let $\triangleleft \in \{<, \leq\}$. For payoff functions* Inf, Sup, LimInf, LimSup, $\underline{\mathsf{MP}}$, $\overline{\mathsf{MP}}$, *and* DS_λ, *determining whether $\mathbf{Reg}_{\mathfrak{S}_\exists, \mathfrak{W}_\forall}(\mathcal{G}) \triangleleft r$ for a given weighted automaton $\mathcal{G}$ is* NP*-hard even if λ is not part of the input.* ■

Proof. We give a reduction from the SAT PROBLEM, *i.e.* satisfiability of a CNF formula. The construction presented is based on a proof in [AKL10]. The argument works for any fixed regret threshold, as long as the weights of the self-loops on the trap states are adapted accordingly. However, for concreteness, we

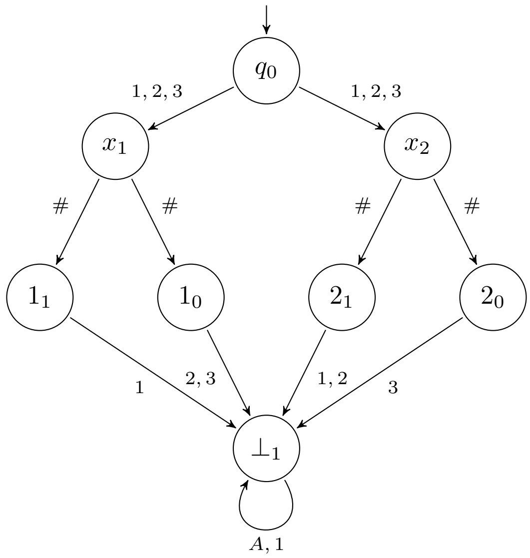

Figure 7.3: Value-choosing gadget for the SAT reduction. Depicted is the configuration for $(x_1 \vee x_2) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$.

will assume a regret threshold of $r = 0$, the Inf payoff function, and consider the non-strict regret threshold problem. The idea of the reduction is simple: given Boolean formula Φ in CNF we construct a weighted automaton $\mathcal{G}_\Phi$ such that Eve can ensure regret value of 0 with a positional strategy in $\mathcal{G}_\Phi$ if and only if Φ is satisfiable. Indeed, the structure of the constructed game forces Eve to play positionally. (And, as we show later, memoryless strategies suffice to ensure 0 regret Lemma 7.1.)

Let us now fix a Boolean formula Φ in CNF with n clauses and m Boolean variables $x_1, \ldots, x_m$. The weighted automaton $\mathcal{G}_\Phi = (Q, q_0, A, \Delta, w)$ has alphabet $A = \{bail, \#\} \cup \{i \mid 1 \leq i \leq n\}$. $\mathcal{G}_\Phi$ includes an initial gadget such as the one depicted in Figure 7.1.

Initial gadget. Here, Eve has the choice of playing left or right. If she plays to the left then Adam can play *bail* and force her to $\bot_0$ while the alternative play resulting from her having chosen to go right goes to $\bot_2$. Hence, playing left already gives Adam a winning strategy to ensure regret 2, so she plays to the right. If Adam now plays *bail* then Eve can go to $\bot_2$ and as $w_{\max} = 2$ this implies the regret will be 0. Therefore, Adam plays anything but *bail*.

Left sub-arena. As the left sub-arena of $\mathcal{G}_\Phi$ we attach the gadget depicted in Figure 7.2. All transitions shown have weight 1 and all missing transitions in order for $\mathcal{G}_\Phi$ to be complete lead to a state $\bot_0$ with a self-loop on every symbol from A with weight 0. Intuitively, as Eve must go to the right sub-arena then all alternative plays in the left sub-arena correspond to either Adam choosing a clause i and spelling $i\#i$ to reach $\bot_1$ or reaching $\bot_0$ by playing any other sequence of symbols.

Right sub-arena. The right sub-arena of the automaton is as shown in Figure 7.3, where all transitions shown have weight 1 and all missing transitions go to $\bot_0$ again. Here, from q_0 we have transitions to state x_j with symbol i if the i-th clause contains variable x_j. For every state x_j we have transitions to j_1 and j_0 with symbol $\#$. The idea is to allow Eve to choose the truth value of x_j. Finally, every state j_1 (or j_0) has a transition to $\bot_1$ with symbol i if the literal x_j (resp. $\neg x_j$) appears in the i-th clause.

The argument to show that Eve can ensure regret of 0 if and only if Φ is satisfiable is straightforward. Assume the formula is indeed satisfiable. Assume, also, that Adam chooses $1 \leq i \leq n$ and spells $i\#i$. Since Φ is satisfiable there is a choice of values for $x_1, \ldots, x_m$ such that for each clause there must be at least one literal in the i-th clause which makes the clause true. Eve transitions, in the right sub-arena from q_0 to the corresponding value and when Adam plays $\#$ she chooses the correct truth value for the variable. Thus, the play reaches $\bot_1$ and, as $w_{\max} = 1$ in Φ it follows that her regret is 0. If Adam does not play as assumed then we know all plays in $\mathcal{G}_\Phi$ reach $\bot_0$ and again her regret is 0. Note that this strategy can be realized with a positional strategy by assigning to each x_j the choice of truth value and choosing from q_0 any valid transition for all $1 \leq i \leq n$.

Conversely, if Φ is not satisfiable then for every valuation of the variables $x_1, \ldots, x_m$ there is at least one clause which is not true. Given any positional strategy of Eve in $\mathcal{G}_\Phi$ we can extract the corresponding valuation of the Boolean variables. Now Adam chooses $1 \leq i \leq n$ such that the i-th clause is not satisfied by the assignment. The play will therefore end in $\bot_0$ while an alternative play in the left sub-arena will reach $\bot_1$. Hence the regret of Eve in the game is 1.

To complete the proof we note that the above analysis is the same for payoff functions Inf, LimInf, LimSup, mean payoff, and discounted sum. For Sup it suffices to change all the weights in the gadgets from 1 to 0.

We observe that, once more, we can adapt the values of the loops in the sinks $\bot_2$, $\bot_1$, and $\bot_0$ to get the same result for the non-strict regret threshold problem and for any regret threshold. For discounted sum, the weights have to be adapted as a function of r and λ. $\square$

Next, for the discounted-sum function we will show that for any fixed λ the general regret threshold problem is PSPACE-hard even if we assume the given automaton has a specific *gap property.*

Theorem 7.2. *Let $\lhd \in \{<, \leq\}$. For payoff function DS_λ, determining whether $\mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{W}_\forall}(\mathcal{G}) \lhd r$ for a given $\delta \in [0,1)$, regret threshold $r \in \mathbb{Q}$, and weighted automaton $\mathcal{G}$ such that either $\mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{W}_\forall}(\mathcal{G}) \lhd r$ or $r + \delta \not\lhd \mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{W}_\forall}(\mathcal{G})$, is* PSPACE*-hard even if λ is not part of the input.* $\blacksquare$

Proof. Given an instance of the QBF problem—that is, a quantified Boolean formula—we construct, in polynomial time, a weighted automaton such that the answer to the regret threshold problem is positive if, and only if, the QBF is true. The main idea behind our reduction is to build an automaton with two disconnected sub-arenas joined by an initial gadget in which we force Eve to go into a specific sub-arena. In order for her to ensure the regret is not too high she must now make sure all alternative plays in the other part of the arena do not achieve too high values. In the sub-arena where Eve finds herself, we will simulate the choice of values for the Boolean variables from the QBF while in the other sub-arena these choices will affect which alternative paths can achieve

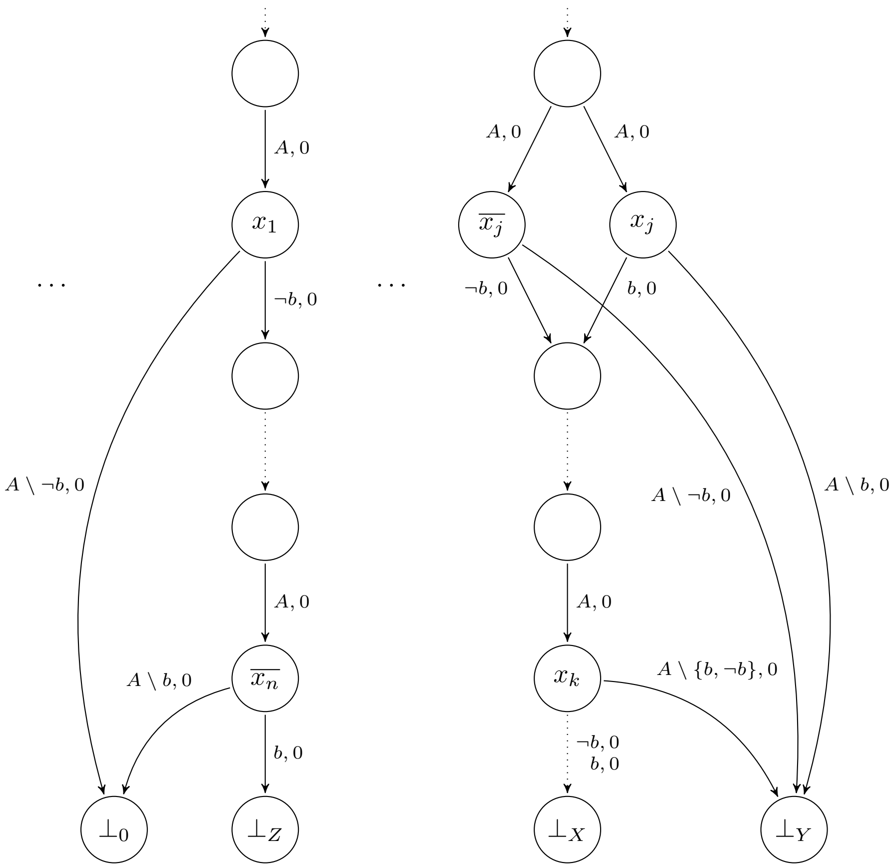

Figure 7.4: Left and right sub-arenas of the reduction from QBF. Clause i shown on the left; existential and universal gadgets for variables x_j and x_k, respectively, on the right.

high discounted-sum values based on the clauses of the QBF. We describe the reduction for $\lhd$ equal to $\leq$. It will be clear how to extend the result to the strict version.

Let us start by fixing an instance of the QBF problem:

$$\exists x_0 \forall x_1 \exists x_2 \ldots \Phi(x_0, x_1, \ldots, x_n)$$

where Φ is in 3-CNF.

Our reduction works for values of positive rationals r, X, Y, and Z such that

(i) $\lambda^2 \frac{Z}{1-\lambda} > r + \delta$,

(ii) $\lambda^{2n} \frac{Z}{1-\lambda} - \lambda^{2n} \frac{X}{1-\lambda} > r + \delta$,

(iii) $\lambda^{2n} \frac{Z}{1-\lambda} - \lambda^{2n} \frac{Y}{1-\lambda} \leq r$,

(iv) $\lambda^3 \frac{Y}{1-\lambda} - \lambda^{2n} \frac{X}{1-\lambda} \leq r$.

The argument works for the strict version of the regret threshold problem if we change the strictness of the above inequalities.

The alphabet of the new weighted arena is $A = \{bail, b, \neg b\}$.

Example assignment. In order to convince the reader that values which satisfy the above inequalities indeed exist for all possible valuations of n and δ we give such a valuation. Let $f : \mathbb{Q} \to \mathbb{Q}$ be defined as $f(x) := \frac{(1-\lambda)x}{\lambda^{2n}}$. Note that, w.l.o.g., we can assume that $n \geq 2$. Consider the valuation

- $r := \lambda^{3-2n}(1 + \delta)$,
- $Z := f(r + \delta + 2)$,
- $X := f(1)$,
- $Y := f(2 + \delta)$.

Clearly, inequalities (i)–(iii) hold. Regarding (iv), it will be useful to consider the equivalent inequality

$$\lambda^{3-2n} Y - X \leq \frac{r(1-\lambda)}{\lambda^{2n}}.$$

We observe that the LHS is smaller than $\lambda^{3-2n}(Y - X)$. Furthermore the difference $Y - X$ is equivalent to $\frac{(1+\delta)(1-\lambda)}{\lambda^{2n}}$. Finally, by choice of r we have that the RHS is equivalent to

$$\lambda^{3-2n} \left(\frac{(1+\delta)(1-\lambda)}{\lambda^{2n}} \right).$$

Hence, (iv) holds as well. Note that the chosen values can be encoded into a polynomial number of bits w.r.t. λ and n as well as the size of the representation of δ.

Initial gadget. The weighted arena we construct starts as is shown in Figure 7.1. Recall that this gadget forces Eve to go into the right sub-arena.

Choosing values. For each existentially quantified variable x_i we will create a "diamond gadget" to allow Eve to choose a different state depending on the value she wants to assign to x_i. From the corresponding states, Adam will have to play b or $\neg b$, respectively, otherwise he allows her to get to $\bot_Y$. For universally quantified variables we have a 2-transition path which allows Adam to choose b or $\neg b$ (in the second step). The right path shown in Figure 7.4 depicts this construction. From (iii) it follows that if Adam cheats at any point during this simulation of value choosing phase of the QBF game, then the play reaches $\bot_Y$ and the regret is at most r. Hence, we can assume that Adam does not cheat and the play eventually reaches $\bot_X$. Observe that the choice of values in this gadget is made as follows: at turn $2i$ after having entered the gadget, the value of x_i is decided.

Clause gadgets. For every clause from Φ we add a path in the new weighted automaton such that every literal ℓ_i in the clause is synchronized with the turn at which the value of x_i is decided in the value-choosing gadget. That is to say, there are $2i-1$ states that must be visited before arriving at the state corresponding to ℓ_i. At state ℓ_i, if the value of x_i corresponding to literal ℓ_i is chosen, the play deterministically goes to $\bot_0$. Otherwise, traversal of the clause-path continues.

It should be clear that if the QBF is true, then Eve has a value-choosing strategy such that at least one literal from every clause holds. That means that every alternative play in the left sub-arena of our construction has been forced into $\bot_0$ while Eve has ensured a discounted-sum value of $\lambda^{2n}\frac{X}{1-\lambda}$ by reaching $\bot_X$. From (iv) it follows that Eve has ensured a regret of at most r. Conversely, if Adam has a value-choosing strategy in the QBF problem so the QBF is show to be false, then he can use his strategy in the constructed arena so that some alternative path in the left sub-arena eventually reaches $\bot_Z$. In this case, from (ii) we get that the regret value is greater than $r+\delta$, as expected. □

We now show EXPTIME-hardness for the payoff functions Inf, Sup, LimInf, LimSup, and mean-payoff, by giving a reduction from *countdown games* [JSL08]. A countdown game $\mathcal{C}$ consists of a weighted graph (S,T), where S is the set of states and $T \subseteq S \times (\mathbb{N}\setminus\{0\}) \times S$ is the transition relation, and a target value $N \in \mathbb{N}$. If $t=(s,d,s') \in T$ then we say that the duration of the transition t is d. A configuration of a countdown game is a pair (s,c), where $s \in S$ is a state and $c \in \mathbb{N}$. A move of a countdown game from a configuration (s,c) consists in player Counter choosing a duration d such that $(s,d,s') \in T$ for some $s' \in S$ followed by player Spoiler choosing s'' such that $(s,d,s'') \in T$, the new configuration is then $(s'',c+d)$. Counter wins if the game reaches a configuration of the form (s,N) and Spoiler wins if the game reaches a configuration (s,c) such that $c < N$ and for all $t=(s,d,\cdot) \in T$ we have that $c+d > N$.

Deciding the winner in a countdown game $\mathcal{C}$ from a configuration $(s,0)$—where N and all durations in $\mathcal{C}$ are given in binary—is EXPTIME-complete. We will show that, given a countdown game, we can construct a game where Eve ensures regret less than 2 if and only if Counter wins in the original countdown game.

Theorem 7.3. *Let $\lhd \in \{<,\leq\}$. For payoff functions* Inf, Sup, LimInf, LimSup,

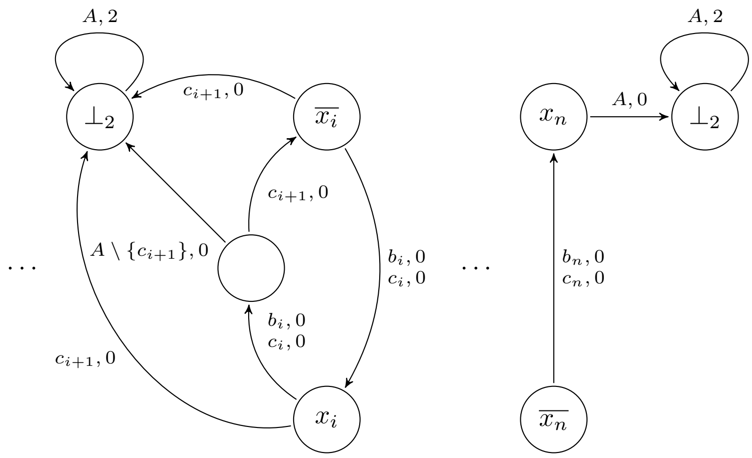

Figure 7.5: Binary counter gadget.

$\underline{\mathsf{MP}}$*, and* $\overline{\mathsf{MP}}$*, determining whether* $\mathbf{Reg}_{\mathfrak{S}_\exists, \mathfrak{W}_\forall}(\mathcal{G}) \lhd r$ *for a given weighted automaton* $\mathcal{G}$ *and regret threshold* $r \in \mathbb{Q}$ *is* EXPTIME*-hard.* ■

Proof of Theorem 7.3. Let us fix a countdown game $\mathcal{C} = ((S,T), N)$ and let $n = \lfloor \log_2 N \rfloor + 2$.

Simplifying assumptions. Clearly, if Spoiler has a winning strategy and the game continues beyond his winning the game, then eventually a configuration (s, c), such that $c \geq 2^n$, is reached. Thus, we can assume w.l.o.g. that plays in $\mathcal{C}$ which visit a configuration (s, N) are winning for Counter and plays which don't visit a configuration (s, N) but eventually get to a configuration (s', c) such that $c \geq 2^n$ are winning for Spoiler.

Additionally, we can also assume that T in $\mathcal{C}$ is total. That is to say, for all $s \in S$ there is some duration d such that $(s, d, s') \in T$ for some $s' \in S$. If this were not the case then for every s with no outgoing transitions we could add a transition $(s, N + 1, s_\perp)$ where $s_\perp$ is a newly added state. It is easy to see that either player has a winning strategy in this new game if and only if he has a winning strategy in the original game.

Reduction. We will now construct a weighted arena $\mathcal{G}$ with $w_{\max} = 2$ such that, in a regret game with payoff function Sup played on $\mathcal{G}$, Eve can ensure regret value strictly less than 2 if and only if Counter has a winning strategy in $\mathcal{C}$.

As all weights are 0 in the arena we build, with the exception of self-loops on sinks, the result holds for Sup, LimSup, Inf, and mean payoff, and the strict threshold problem. We describe the changes required for the Inf and non-strict problems at the end.

Implementation. The alphabet of the weighted arena $\mathcal{G} = (Q, q_0, A, \Delta, w)$ is $A = \{b_i \mid 0 \leq i \leq n\} \cup \{c_i \mid 0 < i \leq n\} \cup \{\mathit{bail}, \mathit{choose}\} \cup S$. We now describe the structure of $\mathcal{G}$ (*i.e.* Q, Δ and w).

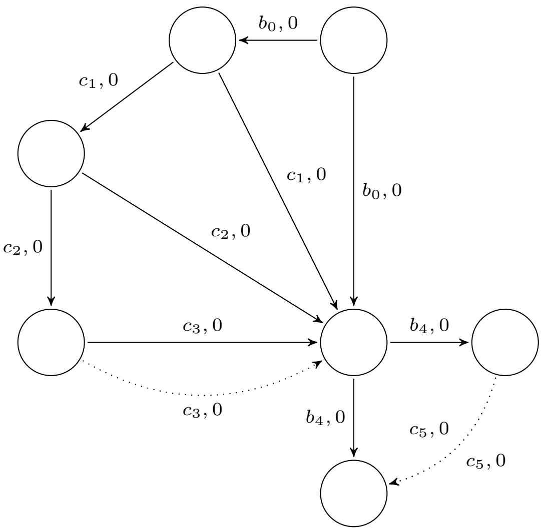

Figure 7.6: Adder gadget: depicted +9.

Initial gadget. Figure 7.1 depicts the initial state of the arena. Once more, we remind the reader that this gadget forces Eve to play into the right sub-arena.

Counter gadget. Figure 7.5 shows the left sub-arena. All states from $\{\overline{x_i} \,|\, 0 \leq i \leq n\}$ have incoming transitions from the left part of the initial gadget with symbol $A \setminus \{bail\}$ and weight 0. Let $y_0 \dots y_n \in \mathbb{B}$ be the "little-endian" binary representation of N. In other words, $(y_i)_0^n$ is such that

$$\sum_0^n 2^i y_i = N.$$

For all x_i such that $y_i = 1$ there is a transition from x_i to $\perp_0$ with weight 0 and symbol *bail*. Similarly, for all $\overline{x_i}$ such that $y_i = 0$ there is a transition from $\overline{x_i}$ to $\perp_0$ with weight 0 and symbol *bail*. All the remaining transitions not shown in the figure cycle on the same state, *e.g.* x_i goes to x_i with symbol *choose* and weight 0.

The sub-arena we have just described corresponds to a counter gadget which keeps track of the sum of the durations "spelled" by Adam. At any point in time, the states of this sub-arena in which Eve believes alternative plays are now will represent the binary encoding of the current sum of durations. Indeed, the initial gadget makes sure Eve plays into the right sub-arena and therefore she knows there are alternative play prefixes that could be at any of the $\overline{x_i}$ states. This corresponds to the 0 value of the initial configuration.

Adder gadget. Let us now focus on the right sub-arena in which Eve finds herself at the moment. The right transition with symbol $A \setminus \{bail\}$ from the initial gadget goes to state s—the initial state from $\mathcal{C}$. It is easy to see how we can simulate Counter's choice of duration and Spoiler's choice of successor. From s there are transitions to every (s, c), such that $(s, c, s') \in T$ for some $s' \in S$ in $\mathcal{C}$, with symbol *choose* and weight 0. Transitions with all other symbols and weight 0 going to $\bot_1$—a sink with a 1-weight cycle with every symbol—from s ensure Adam plays *choose*, lest since $w_{\max} = 2$ the regret of the game will be at most 1 and Eve wins.

Figure 7.6 shows how Eve forces Adam to "spell" the duration c of a transition of $\mathcal{C}$ from (s, c). For concreteness, assume that Eve has chosen duration 9. The top source in Figure 7.6 is therefore the state $(s, 9)$. Again, transitions with all the symbols not depicted go to $\bot_1$ with weight 0 are added for all states except for the bottom sink. Hence, Adam will play b_0 and Eve has the choice of going straight down or moving to a state where Adam is forced to play c_1. Recall from the description of the counter gadget that the belief of Eve encodes the binary representation of the current sum of delays. If she believes a play is in x_1 (and therefore none in $\overline{x_1}$) then after Adam plays b_0 it is important for her to make him play c_1 or this alternative play will end up in $\bot_2$. It will be clear from the construction that Adam always has a strategy to keep the play in the right sub-arena without reaching $\bot_1$ and therefore if any alternative play from the left sub-arena is able to reach $\bot_2$ then Adam wins (*i.e.* can ensure regret 2). Thus, Eve decides to force Adam to play c_1. As the duration was 9 this gadget now forces Adam to play b_4 and again presents the choice of forcing Adam to play c_5 to Eve. Clearly this can be generalized for any duration. This gadget in fact simulates a *cascade configuration* of n 1-bit adders.

Finally, from the bottom sink in the adder gadget, we have transitions with symbols from S with weight 0 to the corresponding states (thus simulating Spoiler's choice of successor state). Additionally, with any symbol from S and with weight 0 Eve can also choose to go to a state q_{bail} where Adam is forced to play *bail* and Eve is forced into $\bot_0$.

Correctness. Note that if the simulation of the counter has been faithful and the belief of Eve encodes the value N then by playing *bail*, Adam forces all of the alternative plays in the left sub-arena into the $\bot_0$ sink. Hence, if Counter has a winning strategy and Eve faithfully simulates the $\mathcal{C}$ she can force this outcome of all plays going to $\bot_0$. Note that from the right sub-arena we have that $\bot_2$ is not reachable and therefore the highest payoff achievable was 1. Therefore, her regret is of at most 1.

Conversely, if both players faithfully simulate $\mathcal{C}$ and the configuration N is never reached, *i.e.* Spoiler had a winning strategy in $\mathcal{C}$ then eventually some alternative play in the left sub-arena will reach x_n and from there it will go to $\bot_2$. Again, the construction makes sure that Adam always has a strategy to keep the play in the right sub-arena from reaching $\bot_1$ and therefore this outcome yields a regret of 2 for Eve.

Changes for Inf. For the same reduction to work for the Inf payoff function we add an additional symbol *kick* to the alphabet of $\mathcal{G}$. We also add deterministic transitions with *kick*, from all states which are not sinks $\bot_x$ for some x, to $\bot_0$. Finally, all non-loop transitions in the initial gadget are now given a weight of 2; the ones in the counter gadget are given a weight of 2 as well; the ones in the adder gadget (*i.e.* right sub-arena) are given a weight of 1.

We observe that if Counter has a winning strategy in the original game $\mathcal{C}$ then Eve still has a winning strategy in $\mathcal{G}$. The additional symbol *kick* allows Adam to force Eve into a 0-loop but also ensures that all alternative plays also go to $\bot_0$, thus playing *kick* is not beneficial to Adam unless an alternative play is already at $\bot_2$. Conversely, if Spoiler has a winning strategy in $\mathcal{C}$ then Adam has a strategy to allow an alternative play to reach $\bot_2$ while Eve remains in the adder gadget. He can then play *kick* to ensure the payoff of Eve is 0 and achieve a maximal regret of 2.

Once again, we observe that the above reduction can be readily parameterized. That is, we can replace the 2 value, the 1 value and the 0 value from the $\bot_2, \bot_1, \bot_0$ sink self-loops by arbitrary values A, B, C satisfying the following constraints:

- $A > B > C$,
- $A - C \geq r$ so that Eve loses by going left in the initial gadget,
- $A - B < r$ so that she does not lose by faithfully simulating the adder if she has a winning strategy from the countdown game, or in other words: if Adam cheats then $A - B$ is low enough to punish him,
- $B - C < r$ so that she does not regret having faithfully simulated addition, that is, if she plays her winning strategy from the countdown game then she does not consider $B - C$ too high and regret it.

Changing the strictness of the last three constraints and finding a suitable valuation for r and A, B, C suffices for the reduction to work for the non-strict regret threshold problem. Such a valuation is given by $A = 2$, $B = 1$, $C = 0$ with $r = 1$. $\square$

To show undecidability of the problem for the mean-payoff function we give a reduction from the threshold problem for *mean-payoff games with partial observation*. This problem was shown to be undecidable in [DDG$^+$10, HPR14]. (We will recall these results in the second part of this dissertation.)

Theorem 7.4. *Let $\lhd \in \{<, \leq\}$. For payoff functions $\underline{\mathsf{MP}}$ and $\overline{\mathsf{MP}}$, determining whether $\mathbf{Reg}_{\mathfrak{S}_\exists, \mathfrak{W}_\forall}(\mathcal{G}) \lhd r$ for a given weighted automaton $\mathcal{G}$ and regret threshold $r \in \mathbb{Q}$ is undecidable.* $\blacksquare$

We recall the definition of the problem we reduce from. An MPG with partial observation $\mathcal{G}$ is a tuple $(Q, q_I, A, \Delta, w, \mathsf{Obs})$ where Q is a set of states, q_I is the initial state of the game, A is a finite set of actions, $\Delta \subseteq Q \times A \times Q$ is the transition relation, $w : \Delta \to \mathbb{Q}$ is a weight function and $\mathsf{Obs} \subseteq \mathcal{P}(Q)$ is a partition of Q into *observations*. In these games a play is started by placing a token on q_I, Eve then chooses an action from A and Adam resolves non-determinism by choosing a valid successor (w.r.t. Δ). Additionally, Eve does not know which state Adam has chosen as the successor, she is only revealed the observation containing the state. More formally: a *concrete play* in such a game is a sequence $q_0 a_0 q_1 a_1 \ldots \in (Q \cdot A)^\omega$ such that $q_0 = q_I$ and $(q_i, a_i, q_{i+1}) \in \Delta$, for all $i \geq 0$. An *abstract play* is then a sequence $\psi = o_0 a_0 o_1 a_1 \ldots \in (\mathsf{Obs} \cdot A)^\omega$ such that there is some concrete play $\pi = q_0 a_0 q_1 a_1 \ldots$ and $q_i \in o_i$, for all $i \geq 0$; in this case we say that π is a *concretization* of ψ. Strategies of Eve in this game are of the form $\sigma : (\mathsf{Obs} \cdot A)^* \mathsf{Obs} \to A$, that is to say they are *observation-based*. Strategies of Adam are not symmetrical, he is allowed to use the exact state information, *i.e.* his strategies are of the form $\tau : (Q \cdot A)^* \to Q$.

The threshold problem for mean-payoff games is defined as follows. Given $\nu \in \mathbb{Q}$, determining whether Eve has an observation-based strategy such that, for all counter-strategies of Adam, the resulting abstract play has no concretization with mean-payoff value (strictly) less than ν. For convenience, let us denote this problem by **maxMPGPO**$(> \nu)$ and **maxMPGPO**$(\geq \nu)$ when the inequality is strict, and non-strict, respectively. Note that in this case Eve is playing to maximize the mean-payoff value of all concrete runs corresponding to the abstract play being played while Adam is minimizing the same.

It was shown in [DDG+10, HPR14] that both problems are undecidable for $\underline{\mathsf{MP}}$ and for $\overline{\mathsf{MP}}$. That is, determining whether **maxMPGPO**$(> \nu)$ or **maxMPGPO**$(\geq \nu)$ is undecidable regardless of the definition used for the mean-payoff function. Further, if we ask for the existence of finite memory observation-based strategies of Eve only, both definitions ($\underline{\mathsf{MP}}$ and $\overline{\mathsf{MP}}$) coincide and determining if **maxMPGPO**$(\geq \nu)$ remains undecidable [DDG+10].

Consider a given MPG $\mathcal{H} = (Q, q_I, A, \Delta, w, \mathsf{Obs})$, and denote by $\mathcal{H}'$ the game obtained by multiplying by -1 all values assigned by w to the transitions of $\mathcal{H}$. Clearly, we get that the answer to whether **maxMPGPO**$(> \nu)$ (resp. **maxMPGPO**$(\geq \nu)$) in $\mathcal{H}$ is affirmative if and only if in $\mathcal{H}'$ Eve has an observation-based strategy to ensure that against any strategy of Adam, the resulting abstract play is such that all concretizations have mean-payoff value[2] of strictly less than $-\nu$ (resp. $\leq -\nu$). Denote these problems by **minMPGPO**$(< \mu)$ and **minMPGPO**$(\leq \mu)$, respectively. It follows that for any definition of the mean-payoff function, these problems are undecidable.

Simplifying assumptions. We assume, w.l.o.g., that in mean-payoff games with partial observation the transition relation is total. As the weights in mean-payoff games with partial observation can be *shifted and scaled*, we can assume w.l.o.g. that ν is any integer $N > 0$. Furthermore, we can also assume that the mean-payoff value of any concrete play in a game is bounded from below by 0 and from above by M (this can again be achieved by shifting and scaling).[3]

Proof of Theorem 7.4. We give a reduction from the threshold problem of *mean-payoff games with partial observation* that resembles the reduction used for the proof of Theorem 7.3. More specifically, given a mean-payoff game with partial observation $\mathcal{H} = (S, s_I, T, B, c, \mathsf{Obs})$, we construct a weighted automaton $\mathcal{G}_{\mathcal{H}} = (Q, q_I, \Delta, A, w)$ with the same payoff function such that

$$\mathbf{Reg}_{\mathfrak{S}_\exists, \mathfrak{W}_\forall}(\mathcal{G}_{\mathcal{H}}) < R$$

if and only if the answer to **minMPGPO**$(< N)$ is affirmative. The reduction we describe works for any R, N, M, C such that

- $C < R$,
- $\frac{M}{2} - C < R$, and
- $\frac{N}{2} = R$,

[2] Technically, this second mean-payoff value must be computed with the lim inf mean-payoff ($\underline{\mathsf{MP}}$) function if the original game used $\overline{\mathsf{MP}}$, and the lim sup mean-payoff ($\overline{\mathsf{MP}}$) function otherwise.

[3] Note that we must have $0 < N < M$, otherwise the threshold problem for any mean-payoff game in which Eve minimizes is trivial.

for concreteness we consider $R = 2$, $N = 4$, $M = 5$ and $C = 1$.

Let us describe how to construct the weighted arena $\mathcal{G}_{\mathcal{H}}$ from $\mathcal{G}$. The alphabet of $\mathcal{G}_{\mathcal{H}}$ is $A = B \cup \{bail\} \cup \mathsf{Obs}$. The structure of $\mathcal{G}_{\mathcal{H}}$ includes a gadget such as the one depicted in Figure 7.1. Recall from the proof of Lemma 7.3 that this gadget ensures Eve chooses to go to the right sub-arena, lest Adam has a spoiling strategy. As the left sub-arena we have a modified version of $\mathcal{H}$. First, for every state $s \in S$ and every action $b \in B$, we add an intermediate state (s, b) such that when b is played from s the play deterministically goes to (s, b) with weight 0. For any transition (s, b, s') in $\mathcal{H}$ we add a transition in $\mathcal{G}_{\mathcal{H}}$ from (s, b) to s' with action $o_{s'}$ and weight $w(s, b, s')$, where $o_{s'}$ is the observation containing s'. Second, we add transitions from every $s \in S$ to $\perp_C$ for symbol *bail* with weight 0 and from every (s, b) to $\perp_C$ with symbol o if there is no $s' \in o$ such that $(s, b, s') \in T$. The sink $\perp_C$ has, for every symbol $a \in A$, a weight C self-loop. As the right sub-arena we will have states q_b for all $b \in B$. For any such q_b there are transitions with weight 0 and symbol b to q_{obs} and transitions with weight 0 and symbols $A \setminus \{b\}$ to $\perp_C$. From q_{obs} with any symbol from Obs, there are 0-weight transitions to $q_{b'}$ (for any $b' \in B$) and transitions with weight 0 and symbols $A \setminus \mathsf{Obs}$ to $\perp_C$. All q_b have incoming edges from the state of the initial gadget which leads to the right sub-arena.

We claim that Eve has a strategy σ in $\mathcal{G}_{\mathcal{H}}$ to ensure regret less than R if and only if the answer to **minMPGPO**$(< N)$ is affirmative. Assume that the latter is the case, *i.e.* in $\mathcal{H}$ Eve has an observation-based strategy to ensure that against any strategy of Adam the abstract play has no concretization with mean-payoff value greater than or equal to N. Let us describe the strategy of Eve in $\mathcal{G}_{\mathcal{H}}$. First, she plays into the right sub-arena of the game. Once there, she tries to visit states $q_{b_0} q_{b_1} \ldots$ based on her strategy for $\mathcal{H}$. If Adam, at some q_{b_i} does not play b_i, or at some visit to q_{obs} he plays a non-observation symbol, then Eve goes to $\perp_C$. The play then has value C. Since no alternative play in the left sub-arena can have value greater than $\frac{M}{2}$ and we have that $\frac{M}{2} - C < R$, Eve wins. Thus, we can assume that Adam, at every q_{b_i} plays the symbol b_i and at every visit to q_{obs} plays an observation. Note that, by construction of the left sub-arena, we are forcing Adam to reveal a sequence of observations to Eve and allowing her to choose a next action. It follows that the value of the play in $\mathcal{G}_{\mathcal{H}}$ is 0. Any alternative play in the right sub-arena would have value of at most C as the highest weight in it is C. In the left sub-arena, we have that all alternative plays have value strictly less than $\frac{N}{2}$. Indeed, since she has followed her winning strategy from $\mathcal{H}$, and since by construction we have that all alternative plays in the left sub-arena correspond to concretizations of the abstract path spelled by Adam and Eve, if there were some play with value of at least $\frac{N}{2}$ this would contradict her strategy being optimal. As $C < R$ and $\frac{N}{2} = R$, we have that Eve wins the regret game, *i.e.* her strategy ensures regret strictly less than R.

Conversely, assume that the answer to **minMPGPO**$(< N)$ is negative. Then regardless, of which strategy from $\mathcal{H}$ Eve decides to follow, we know there will be some alternative play in the left sub-arena with value of at least $\frac{N}{2}$. If Adam allows Eve to play any such strategy then the value of the play is 0 and her regret is at least $\frac{N}{2} = R$, which concludes the proof for the strict regret threshold problem.

We observe that the restrictions on N, M, R and C can easily be adapted to allow for a reduction from **minMPGPO**($\leq N$) to the non-strict regret threshold problem. □

We observe that in the construction used in the above proof Eve might require infinite memory as it is known that, in mean-payoff games with partial observation, the protagonist might require infinite memory to win. Yet, as we have already mentioned, even if we ask whether Eve has a winning finite memory observation-based strategy, the **maxMPGPO**($\geq \nu$) problem—and thus the **minMPGPO**($\leq \mu$) problem—remains undecidable. Notice that the above construction, when restricting Eve to play with finite memory, gives us a reduction from this exact problem. Hence, even when restricting Eve to use only finite memory, the problem is undecidable.

Proposition 7.3. *For payoff functions* $\underline{\mathsf{MP}}$ *and* $\overline{\mathsf{MP}}$*, determining whether*

$$\mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{W}_\forall}(\mathcal{G}) \leq r$$

for a given weighted automaton $\mathcal{G}$ *and regret threshold* $r \in \mathbb{Q}$ *is undecidable even if Eve is only allowed to play finite memory strategies.* ■

7.2 Upper Bound for 0-Regret

In this section, our goal is to convince the reader about two facts. First, there is a strategy for Eve to ensure regret 0 if and only if the regret value of an automaton, or game, is 0. (Recall that the definition of the regret of a game allows for regret 0 even if no particular strategy for Eve witnesses regret 0—instead, a sequence of strategies might have regret approaching 0 in the limit.) Second, deciding if Eve has a regret-free strategy is in NP for all payoff functions. The latter is optimal in view of Theorem 7.1.

7.2.1 Existence of regret-free strategies

We will now show that if the regret of an arena (or automaton) is 0, then we can construct a memoryless strategy for Eve which ensures no regret is incurred. More specifically, assuming the regret is 0, we have the existence of a family of strategies of Eve which ensure decreasing regret (with limit 0). We use this fact to choose a small enough ε and the corresponding strategy of hers from the aforementioned family to construct a memoryless strategy for Eve with nice properties which allow us to conclude that its regret is 0.

Lemma 7.1. *For payoff functions* Inf, Sup, LimInf, LimSup, $\underline{\mathsf{MP}}$, $\overline{\mathsf{MP}}$, *and* DS_λ, *in any weighted automaton* $\mathcal{G}$ *we have that* $\mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{W}_\forall}(\mathcal{G}) = 0$ *if and only if there exists* $\sigma \in \mathfrak{S}_\exists^1$ *such that* $\mathbf{reg}^\sigma_{\mathfrak{S}_\exists,\mathfrak{W}_\forall}(\mathcal{G}) = 0$. ■

Proof. Clearly if regret-free strategies exist, then the regret of the automaton is 0. Hence, we concentrate on the other implication.

Note that for a given automaton $\mathcal{G} = (Q, q_0, A, \Delta, w)$ with payoff function Inf, Sup, LimInf, or LimSup, we have that the set of possible regret values for the automaton is finite. Indeed, the regret value must be an element from

$$\{|w(d) - w(d')| : d, d' \in \Delta\}$$

since all runs will have a value from $\{w(d) : d \in \Delta\}$. It follows that there is a strategy for Eve which ensures regret at most r if and only if the regret of the automaton is at most r.

We now claim that any regret-free strategy σ with the latter property can be turned into a positional strategy for Eve which also ensures regret 0.[4] Consider the (possibly infinite) Mealy machine $(M, m_0, \alpha_u, \alpha_o)$ realizing σ and its product with the automaton: $\mathcal{G} \times \sigma$. The strategy σ can be turned into a positional one if for all states $q \in Q$, for all pairs (q, m) and (q, m') from $\mathcal{G} \times \sigma$ there is one of the two which is always 'better'. More formally, (q, m) is better than (q, m') if σ, with internal memory state m, reads from q any infinite word β and constructs a run with a strictly a higher value than it does if it has internal memory state m'. Clearly, if (q, m) is better than (q, m') we can redirect all transitions leading to (q, m') so that they now go to (q, m) and the strategy will use less memory (at least for state q). If this process cannot be repeated to get a positional strategy, then at some point we must have obtained a strategy σ' with two memory elements m, m' such that for some state q and for some infinite word β, the value of the run built by σ from q and memory element m is strictly better than the one built starting with memory element m and for some infinite word β' the opposite is true. However, this is a contradiction. Indeed, (q, m) and (q, m') are reachable from q_I via different runs on different word prefixes κ, κ' respectively. Nonetheless, q in $\mathcal{G}$ is reachable via runs on both prefixes. Hence, σ' and therefore σ is cannot be 0-regret if it constructs a run for word $\kappa \cdot \beta'$ (and for $\kappa' \cdot \beta$) which is not optimal in $\mathcal{G}$. It follows that if $\mathcal{G}$ has regret value 0 then Eve has a memoryless regret-free strategy.

This leaves us with mean payoff and discounted sum.

Discounted Sum. Consider a fixed weighted automaton $\mathcal{A} = (Q, q_0, A, \Delta, w)$ and a discount factor $\lambda \in (0, 1)$. Further, we suppose the regret of $\mathcal{A}$ is 0.

Let us start by defining a set of values which represent lower bounds on the regret Eve can get by resolving the non-determinism of $\mathcal{A}$ on the fly. First, let us introduce some additional notation. Define $\mathcal{A}^q := (Q, q, A, \Delta, w)$, *i.e.* the automaton $\mathcal{A}$ with new initial state q. For states $q, q' \in Q$, let $\mu(q, q') := \sup\left(\{\mathcal{A}^{q'}(\alpha) - \mathcal{A}^q(\alpha) \mid \alpha \in A^\omega\} \cup \{0\}\right)$. We now let M be equal to the set of values:

$$\{|w(p, a, q') - w(p, a, q) + \lambda \cdot \mu(q, q')| : p \in Q \text{ and } q, q' \text{ are } a\text{-successors of } p\}.$$

Note that since $\mathcal{A}$ is assumed to be total (*i.e.*, every state-action pair has at least one successor) then M cannot be empty. Observe that, by definition, M only contains non-negative values. Since $\mathcal{A}$ has regret 0, then we know that for all $\delta \in (0, 1)$, there is a strategy σ_δ of Eve such that $\mathbf{reg}^{\sigma_\delta}_{\mathfrak{S}_\exists, \mathfrak{W}_\forall}(\mathcal{A}) \le \delta$. If $M \neq \{0\}$, we let $0 < \varepsilon < \lambda^{|Q|} \cdot (\min M \setminus \{0\})$. We now define a memoryless strategy σ of Eve as follows: if $M = \{0\}$ then σ is arbitrary, otherwise $\sigma(p, a) = q$ implies that there is some path $\tau = q_0 \dots q_n$ such that $q_n = p$, $n < |Q|$, and $\tau \cdot a \cdot q$ is consistent with σ_ε. It is easy to show that for all play prefixes, for all actions, σ chooses a successor which ensures no other successor can read a word and obtain a "better" value. Formally,

[4] The argument we present is in fact an adaptation of one given in [AKL10].

Claim 12. *For all* $\pi = q_0 a_0 \dots q_n$ *and all* $a \in A$, *if* $\sigma(\pi, a) = q$ *then for all* $q' \in \mathsf{post}_a(q_n)$ *the following holds*

$$\forall \alpha \in A^\omega : w(p, a, q') + \lambda \mathcal{A}^{q'}(\alpha) \leq w(p, a, q) + \lambda \mathcal{A}^{q}(\alpha).$$

■

We will see later (in the proof of Theorem 7.5) that, given a finite memory strategy for Eve, we can reduce the question of whether the strategy is regret-free to determining the winner of a one-player discounted-sum game. The game is played on the product of $\mathcal{A}$ restricted to the fixed strategy and $\mathcal{A}$. Using the above claim, we can use the same argument as presented in the previous chapters. That is, if Adam can read some word in a "better" way from some other state than the one chosen by σ, then by the above claim, making some changes to his strategy, we can show that him using σ also has the property of being "better" and we thus get a contradiction. (See the proofs of Lemma 5.1 and Lemma 6.5. Also note that it suffices to consider ultimately periodic words and runs since Adam can be assumed to play positionally in the constructed game.)

Mean payoff. For mean payoff we follow a similar idea as for discounted sum. The main difference is that we must set M to be

$$\{\mu(q, q') : \exists (p, a) \in Q \times A \text{ and } q, q' \text{ are } a\text{-successors of } p\}.$$

If $M \neq \{0\}$ we choose $\varepsilon < \min(M \setminus \{0\})$. We now select some σ_ε which ensures regret at most ε and define σ as for discounted sum. If $M = \{0\}$ then we let σ be any memoryless strategy. In this context, it is easy to see that

Claim 13. *For all* $\pi = q_0 a_0 \dots q_n$ *and all* $a \in A$, *if* $\sigma(\pi, a) = q$ *then for all* $q' \in \mathsf{post}_a(q_n)$ *the following holds*

$$\forall \alpha \in A^\omega : \mathcal{A}^{q'}(\alpha) \leq \mathcal{A}^{q}(\alpha).$$

■

We then, essentially, construct a new mean-payoff game in which σ is fixed as the strategy for Eve and Adam tries to find a word and a "better" way to read the word than that prescribed by σ. If we assume such a strategy for him exists, we can derive a contradiction using the above claim. (Observe that here the fact that Adam plays positionally is not obvious. It will however be shown to hold in Lemma 7.2.) □

7.2.2 Regular words suffice for Adam

In this section we will argue that, if Eve plays a finite memory strategy, then the regret of that strategy is witnessed by an *ultimately periodic* word played by Adam. This will later allow us to give a simple proof of the 0-regret threshold problem being in NP.[5] The main idea of the latter is to guess the strategy of Eve, which will be positional according to the results from the previous section, and consider the product of the restriction of the automaton—to the

[5]We will actually prove a more general result with 0-regret being a particular sub-case.

chosen strategy—with the original automaton. The transitions of the resulting product automaton are then weighted with the difference of the corresponding weights from the restricted, deterministic, automaton and the original one. Now, the question of whether the strategy has regret r corresponds to testing the non-emptiness of the automaton with accepting condition defined by a strict threshold r. It is not immediate, at first, that the construction is correct. Indeed, the limit of the differences of weights yielding two mean-payoff values, for instance, is not necessarily the difference of the two values (limits themselves). However, when the automaton is finite and the word is regular, this is the case.

Theorem 7.5. *Consider a positive integer $m \in \mathbb{N}_{>0}$ and let $\lhd \in \{<, \leq\}$. For payoff functions* Inf*,* Sup*,* LimInf*,* LimSup*,* $\underline{\mathsf{MP}}$*,* $\overline{\mathsf{MP}}$*, and* DS_λ*, determining whether $\mathbf{Reg}_{\mathfrak{S}_\exists^m,\mathfrak{W}_\forall}(\mathcal{G}) \lhd r$, for a given weighted automaton $\mathcal{G}$ and regret threshold $r \in \mathbb{Q}$ can be done in* $\mathrm{NTIME}(m^2|\mathcal{G}|^2)$ *even if λ is part of the input.* ■

Denote by $\mathfrak{R}_\forall \subseteq \mathfrak{W}_\forall$ the set of all word strategies of Adam which are regular. That is to say, $\alpha \in \mathfrak{R}_\forall$ if and only if α is *ultimately periodic*:

$$\alpha = \beta \cdot \kappa^\omega,$$

where β and κ are both finite words. It is well-known that the mean-payoff value of ultimately periodic plays in weighted arenas is the same for both $\overline{\mathsf{MP}}$ and $\underline{\mathsf{MP}}$.

Before proving the theorem we first show that ultimately periodic words suffice for Adam to spoil a finite memory strategy of Eve. Let us fix some useful notation. Given weighted automaton $\mathcal{G}$ and a finite memory strategy σ for Eve in $\mathcal{G}$ we denote by $\mathcal{G}_\sigma$ the deterministic automaton embodied by a refinement of $\mathcal{G}$ that is induced by σ.

Lemma 7.2. *For $r \in \mathbb{Q}$, weighted automaton $\mathcal{G}$, and payoff functions* $\underline{\mathsf{MP}}$ *and* $\overline{\mathsf{MP}}$*, if $\mathbf{Reg}_{\mathfrak{S}_\exists^m,\mathfrak{W}_\forall}(\mathcal{G}) \rhd r$ then $\mathbf{Reg}_{\mathfrak{S}_\exists^m,\mathfrak{R}_\forall}(\mathcal{G}) \rhd r$, for $\rhd \in \{>, \geq\}$.* ■

Proof. We prove the claim for $\underline{\mathsf{MP}}$ and $\geq$ but the result for $\overline{\mathsf{MP}}$ follows from minimal changes to the argument (a small quantifier swap in fact) and for $>$ variations we need only use the strict versions of Equations (7.1) and (7.2). We assume without loss of generality that all weights are non-negative.

Let σ be a strategy for Eve in $\mathcal{G}$ which uses at most memory m. We claim that if Adam has a word strategy to ensure the regret of such a strategy for Eve in $\mathcal{G}$ is at least r, then he also has a regular word strategy to do so.

Consider the bi-weighted graph $\mathcal{G}$ constructed by taking the synchronous product of $\mathcal{G}$ and $\mathcal{G}_\sigma$ while labelling every edge with two weights: the value assigned to the transition by the weight function of $\mathcal{G}_\sigma$ and the value assigned to the transition by that of $\mathcal{G}$. For a path π in $\mathcal{G}$, denote by $w_i(\pi)$ the sum of the weights of the edges traversed by π w.r.t. the i-th weight function. Also, for an infinite path π, denote by $\underline{\mathsf{MP}}_i$ the mean-payoff value of π w.r.t. the i-th weight function. Clearly, Adam has a word strategy to ensure a regret of at least r against the strategy σ of Eve if and only if there is an infinite path π in $\mathcal{G}$ such that $\underline{\mathsf{MP}}_2(\pi) - \underline{\mathsf{MP}}_1(\pi) \geq r$. We claim that if this is the case then there is a simple cycle χ in $\mathcal{G}$ such that $\frac{1}{|\chi|}w_2(\chi) - \frac{1}{|\chi|}w_1(\chi) \geq r$. The argument is based on the cycle decomposition of π (see, *e.g.* [EM79]).

Assume, for the sake of contradiction, that all the cycles χ in $\mathcal{G}$ satisfy the following:

$$\frac{1}{|\chi|}w_2(\chi) - \frac{1}{|\chi|}w_1(\chi) \leq r - \varepsilon, \text{ for some } 0 < \varepsilon \leq r, \tag{7.1}$$

and let us consider an arbitrary infinite path $\pi = v_0 v_1 \ldots$. Let $l = \underline{\mathsf{MP}}_1(\pi)$. We will show

$$\liminf_{k\to\infty} \frac{w_2(\langle v_j\rangle_{j\leq k})}{k} - l \leq r - \varepsilon, \tag{7.2}$$

from which the required contradiction follows.

For any $k \geq 0$, the cycle decomposition of $\langle v_j\rangle_{j\leq k}$ tells us that apart from a small sub-path, π', of length at most n (the number of states in $\mathcal{G}$), the prefix $\langle v_j\rangle_{j\leq k}$ can be decomposed into simple cycles $\chi_1, \ldots, \chi_t$ such that $w_i(\langle v_j\rangle_{j\leq k}) = w_i(\pi') + \sum_{j=1}^{t} w_i(\chi_j)$ for $i = 1, 2$. If $w_{\max}$ is the maximum weight occurring in $\mathcal{G}$, then from Equation (7.1) we have:

$$\begin{aligned} w_2(\langle v_j\rangle_{j\leq k}) &\leq nw_{\max} + \sum_{j=1}^{t} w_2(\chi_j) \\ &\leq nw_{\max} + (r-\varepsilon)\sum_{j=1}^{t}|\chi_j| + \sum_{j=1}^{t} w_1(\chi_j) \\ &\leq nw_{\max} + k(r-\varepsilon) + w_1(\langle v_j\rangle_{j\leq k}). \end{aligned}$$

Now, it follows from the definition of the limit inferior that for any $\varepsilon' > 0$ and any $K > 0$ there exists $k > K$ such that $w_1(\langle v_j\rangle_{j\leq k}) \leq k(l+\varepsilon')$. Thus for any $\varepsilon' > 0$ and $K' > 0$, there exists $k > \max\{K', nw_{\max}/\varepsilon'\}$ such that

$$\frac{w_2(\langle v_j\rangle_{j\leq k})}{k} \leq \frac{nw_{\max}}{k} + (r-\varepsilon) + (l+\varepsilon') < (l+r-\varepsilon) + 2\varepsilon'.$$

Equation (7.2) then follows from the definition of limit inferior.

The above implies that Adam can, by repeating χ infinitely often, achieve a regret value of at least r against strategy σ of Eve. As this can be done by him playing a regular word, the result follows. ☐

We now proceed with the proof of the theorem. The argument is presented for mean payoff ($\underline{\mathsf{MP}}$) and discounted sum but minimal changes are required for the other payoff functions. For simplicity, we use the non-strict threshold for the emptiness problems. However, the complexity of deciding emptiness of quantitative languages (the result we use from [CDH10]) is independent of this. Further, the exact same argument presented here works for both cases. Thus, it suffices to show the result follows for $\geq$.

Proof of Theorem 7.5. We will "guess" a strategy σ for Eve which uses memory at most m and verify (in polynomial time w.r.t. m and the size of $\mathcal{G}$) that it ensures a regret value of strictly less than r.

For all prefix-independent functions but mean payoff, the result is actually a corollary of the reduction to parity and Streett games used to prove Lemma 7.3. There, the product of the original automaton and a deterministic automaton is considered. Here we shall do the same, however, the deterministic automaton we take is induced by the strategy σ we have guessed for Eve. Here, the resulting product game is actually a one-player game since the choices of Eve have been fixed already. Further, the product game is not exponential (as is the case in the proof of Lemma 7.3). Since emptiness for Rabin—the dual of the Streett objective—and parity automata is decidable in polynomial time, the result follows.

In the sequel we will focus on mean payoff and discounted sum. It will be clear how to generalize to the non-strict version of the problem.

Let $\mathcal{A}$ be the weighted ($\underline{\mathsf{MP}}$) automaton constructed as the synchronous product of $\mathcal{G}$ and $\mathcal{G}_\sigma$. The new weight function maps a transition to the difference of the values of the weight functions of the two original automata. We claim that the language of $\mathcal{A}$ is empty (for accepting threshold $\geq r$) if and only if $\mathbf{reg}^{\sigma}_{\mathfrak{S}^m_\exists,\mathfrak{W}_\forall}(\mathcal{G}) < r$. Indeed, there is a bijective map from every run of $\mathcal{A}$ to a pair of plays π, π' in $\mathcal{G}$ such that both π and π' are consistent with the same word strategy of Adam and π is consistent with σ. It should be clear that $\mathcal{A}$ has size at most $m|\mathcal{G}|$. As emptiness of a weighted automaton $\mathcal{A}$ can be decided in $\mathcal{O}(|\mathcal{A}|^2)$ time [CDH10], the result will follow.

Discounted sum. Since the difference of two convergent series is equivalent to the difference of their limits, every infinite run of $\mathcal{A}$ is assigned exactly the value of the difference of the discounted sum of the corresponding runs π and π' from $\mathcal{G}$. The converse is also true. Hence the result follows.

Mean payoff. We now show that if the language of $\mathcal{A}$ is not empty then Adam can ensure a regret value of at least r against σ in $\mathcal{G}$ and that, conversely, if Adam has a spoiling strategy against σ in $\mathcal{G}$ then that implies the language of $\mathcal{A}$ is not empty.

Let ϱ_x be a run of $\mathcal{A}$ on x. From the definition of $\mathcal{A}$ we get that $\underline{\mathsf{MP}}(\varrho_x) = \liminf_{i\to\infty} \frac{1}{i}\sum_{j=0}^{i}(a_j - b_j)$ where $\alpha_x = \langle a_i\rangle_{i\geq 0}$ and $\beta_x = \langle b_i\rangle_{i\geq 0}$ are the infinite sequences of weights assigned to the transitions of ϱ by the weight functions of $\mathcal{G}$ and $\mathcal{G}_\sigma$ respectively. It is known that if a mean-payoff automaton accepts a word y then it must accept an ultimately periodic word y', thus we can assume that x is ultimately periodic (see, *e.g.* [CDH10]). Furthermore, we can also assume the run of the automaton on x is ultimately periodic. Recall that for ultimately periodic runs we have that $\underline{\mathsf{MP}}(\varrho_x) = \overline{\mathsf{MP}}(\varrho_x)$. We get the following

$$
\begin{aligned}
\underline{\mathsf{MP}}(\varrho_x) &= \limsup_{i\to\infty} \frac{1}{i}\sum_{j=0}^{i}(a_j - b_j) \\
&\leq \limsup_{i\to\infty} \frac{1}{i}\sum_{j=0}^{i} a_j + \limsup_{i\to\infty} \frac{-1}{i}\sum_{j=0}^{i} b_j && \text{sub-additivity of } \limsup \\
&\leq \limsup_{i\to\infty} \frac{1}{i}\sum_{j=0}^{i} a_j - \liminf_{i\to\infty} \frac{1}{i}\sum_{j=0}^{i} b_j \\
&\leq \liminf_{i\to\infty} \frac{1}{i}\sum_{j=0}^{i} a_j - \liminf_{i\to\infty} \frac{1}{i}\sum_{j=0}^{i} b_j && \text{ultimate periodicity.}
\end{aligned}
$$

Thus, as x and ϱ_x can be be mapped to a strategy of Adam in $\mathcal{G}$ which ensures regret of at least r against σ, the claim follows.

For the other direction, assume Adam has a word strategy τ in $\mathcal{G}$ which ensures a regret of at least r against σ. From Lemma 7.2 it follows that τ and the run ϱ of $\mathcal{G}$ with value $\mathcal{G}(\tau)$ can be assumed to be ultimately periodic w.l.o.g.. Denote by ϱ_σ and w_σ the run of $\mathcal{G}_\sigma$ on τ and the weight function of $\mathcal{G}_\sigma$ respectively. We then get that

$$
\begin{aligned}
&\liminf_{i\to\infty} \frac{1}{i} w_\sigma(\varrho_\sigma) - \liminf_{i\to\infty} \frac{1}{i} w(\varrho) \\
&= \liminf_{i\to\infty} \frac{1}{i} w_\sigma(\varrho_\sigma) + \limsup_{i\to\infty} \frac{-1}{i} w(\varrho) \\
&= \liminf_{i\to\infty} \frac{1}{i} w_\sigma(\varrho_\sigma) + \liminf_{i\to\infty} \frac{-1}{i} w(\varrho) && \text{ultimate periodicity} \\
&\leq \underline{\mathsf{MP}}(\psi_\tau) && \text{super-additivity of } \liminf,
\end{aligned}
$$

where ψ_τ is the corresponding run of $\mathcal{A}$ for τ and ϱ. Hence, $\mathcal{A}$ has at least one word in its language. $\square$

7.3 Upper Bounds for Prefix-Independent Functions

In this section, we provide tight upper bounds for all the prefix-independent payoff functions except for mean-payoff: the regret threshold problem is in EXPTIME for Sup, Inf, LimSup, and LimInf.

The following result summarizes the results of this section:

Theorem 7.6. *For payoff functions* Inf*,* Sup*,* LimInf*, and* LimSup*, the regret of a game played against an eloquent adversary can be computed in exponential time.* ■

There is an exponential-time algorithm for solving the regret threshold problem for Inf, Sup, LimInf, and LimSup. This algorithm is obtained by a reduction to parity or Streett games. Since all of these payoff functions have a finite range, one can easily implement a binary search—based on queries to the regret threshold problem solver—to compute the regret of the game. Furthermore, the range of the payoff functions is linear with respect to the size of the given automaton. Hence, it suffices to show the regret threshold problem is solvable in exponential time for the desired result to follow.

Lemma 7.3. *For $r \in \mathbb{Q}$, weighted automaton $\mathcal{G}$ and payoff function* Inf*,* Sup*,* LimInf*, or* LimSup*, determining whether $\mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{W}_\forall}(\mathcal{G}) \lhd r$, for $\lhd \in \{<, \leq\}$, can be done in exponential time.* ■

We show how to decide the strict regret threshold problem. However, the same algorithm can be adapted for the non-strict version by changing strictness of the inequalities used to define the parity/Streett accepting conditions.

Proof. We focus on the LimInf and LimSup payoff functions. The result for Inf and Sup follows from the translation to LimInf and LimSup games given in Section 5.3. Our decision algorithm consists in first building a deterministic automaton for $\mathcal{G} = (Q_1, q_I, A, \Delta_1, w_1)$ using the construction provided in [CDH10]. We denote by $\mathcal{D}_\mathcal{G} = (Q_2, s_I, A, \Delta_2, w_2)$ this deterministic automaton and we know that it is at most exponentially larger than $\mathcal{G}$. Next, we consider a simulation game played by Eve and Adam on the automata $\mathcal{G}$ and $\mathcal{D}_{\mathcal{G}}$. The game is played for an infinite number of rounds and builds runs in the two automata, it starts with the two automata in their respective initial states (q_I, s_I), and if the current states are q_1 and q_2, then the next round is played as follows:

- Adam chooses a letter $a \in A$, and the state of the deterministic automaton is updated accordingly, *i.e.* $q_2' = \Delta_2(q_2, a)$, then

- Eve updates the state of the non-deterministic automaton to q_1' by reading a using one of the edges labelled with a in the current state, *i.e.* she chooses q_1' such that $q_1' \in \Delta_1(q_1, a)$. The new state of the game is (q_1', q_2').

Eve wins the simulation game if the minimal weight seen infinitely often in the run of the non-deterministic automaton is larger than or equal to the minimal weight seen infinitely often in the deterministic automaton minus r. It should be clear that this happens exactly when Eve has a regret bounded by r in the original regret game on the word which is spelled out by Adam.

Let us focus on the $\liminf$ payoff function now. We will sketch how this game can be translated into a parity game. For completeness, we now recall the formal definition of the latter. A parity game is a pair $(\mathcal{G}, \Omega)$ where $\mathcal{G}$ is a non-weighted arena and $\Omega : V \to \mathbb{N}$ is a function that assigns a priority to each vertex. Plays, strategies, and other notions are defined as with games played on weighted arenas. A play in a parity game induces an infinite sequence of priorities. We say a play is winning for Eve if and only if the minimal priority seen infinitely often is odd.[6] The *parity index* of a parity game is the number of priorities labelling its vertices, that is $|\{\Omega(v) \mid v \in V\}|$.

To obtain the translation, we keep the structure of the game as above but we assign priorities to the edges of the games instead of weights. We do it in the following way. If $X = \{x_1, x_2, \dots, x_n\}$ is the ordered set of weight values that appear in the automata (note that $|X|$ is bounded by the number of edges in the non-deterministic automaton), then we need the set of priorities $D = \{2, \dots, 2n+1\}$. We assign priorities to edges in the game as follows:

- when Adam chooses a letter a from q_2, then if the weight that labels the transition that leaves q_2 with letter a in the deterministic automaton is equal to $x_i \in X$, then the priority is set to $2i + 1$,

- when Eve updates the non-deterministic automaton from q_1 with an edge labelled with weight w, then the color is set to $2i$ where i is the index in X such that $x_{i-1} \leq w + r < x_i$.

It should be clear that along a run, the minimal color seen infinitely often is odd if and only if the corresponding run is winning for Eve in the simulation game. So, now it remains to solve a parity game with exponentially many states and polynomially many priorities w.r.t. the size of $\mathcal{G}$. This can be done in exponential time with classical algorithms for parity games.

[6] Equivalently, one could ask for the minimal priority seen infinitely often to be even, just add 1 to all priorities. However, odd priorities make our argument easier to present here.

LimSup to Streett games. Let us now focus on LimSup. In this case we will reduce our problem to that of determining the winner of a Streett game with state-space exponential w.r.t. the original game but with number of Streett pairs polynomial (w.r.t. the original game). Recall that a Streett game is a pair $(G, \mathcal{F})$ where G is a game graph (with no weight function) and $\mathcal{F} \subseteq \mathcal{P}(V) \times \mathcal{P}(V)$ is a set of Streett pairs. We say a play is winning for Eve if and only if for all pairs $(E, F) \in \mathcal{F}$, if a vertex in E is visited infinitely often then some vertex in F is visited infinitely often as well.

Consider a LimSup automaton $\mathcal{G} = (Q, q_I, A, \Delta, w)$. For $x_i \in \{w(d) \mid d \in \Delta\}$ let us denote by $\mathcal{A}^{\geq x_i}$ the Büchi automaton with Büchi transition set equivalent to all transitions with weight of at least x_i. We denote by $\mathcal{D}^{\geq x_i} = (Q_i, q_{i,I}, A, \delta_i, \Omega_i)$ the deterministic parity automaton with the same language as $\mathcal{A}^{\geq x_i}$.[7] From [Pit07] we have that $\mathcal{D}^{\geq x_i}$ has at most $2|Q|^{|Q|}|Q|!$ states and parity index $2|Q|$ (the number of priorities). Now, let $x_1 < x_2 < \cdots < x_l$ be the weights appearing in transitions of $\mathcal{G}$. We construct the (non-weighted) arena $\Gamma_{\mathcal{G}} = (V, V_\exists, E, v_I)$ and Streett pair set $\mathcal{F}$ as follows

- $V = Q \times \prod_{i=1}^{l} Q_i \cup Q \times \prod_{i=1}^{l} Q_i \times A \cup Q \times \prod_{i=1}^{l} Q_i \times A \times Q$;
- $V_\exists = Q \times \prod_{i=1}^{l} Q_i \times A$;
- $v_I = (q_I, q_{1,I}, \dots, q_{l,I})$;
- E contains
  - $\big((p, p_1, \dots, p_l)), (p, p_1, \dots, p_l, a)\big)$ for all $a \in A$,
  - $\big((p, p_l, \dots, p_l, a), (p, p_1, \dots, p_l, a, q)\big)$ if $(p, a, q) \in \Delta$,
  - $\big((p, p_l, \dots, p_l, a, q), (q, q_1, \dots, q_l)\big)$ if for all $1 \leq i \leq l$: $(p_i, a, q_i) \in \delta_i$;
- For all $1 \leq i \leq l$ and all even y such that $\mathsf{Range}(\Omega_i) \ni y$, $\mathcal{F}$ contains the pair (E_i, F_i) where
  - $E_{i,y} = \{(p, \dots, p_i, \dots, p_l, a, q) \mid \Omega_i(p_i, a, \delta(p_i, a)) = y\}$, and
  - $F_{i,y} = \{(p, \dots, p_j, \dots, p_l, a, q) \mid (\Omega_i(p_i, a, \delta(p_i, a)) < y \wedge y \pmod 2 = 1) \vee w(p, a, q) \geq x_i - r\}$.

It is not hard to show that in the resulting Streett game, a strategy σ of Eve is winning against any strategy τ of Adam if and only if for every automaton $\mathcal{D}^{\geq x_i}$ which accepts the word induced by τ then the run of $\mathcal{G}$ induced by σ has payoff of at least $x_i - r$, if and only if Eve has a winning strategy in $\mathcal{G}$ to ensure regret is less than r.

Note that the number of Streett pairs in $\Gamma_{\mathcal{G}}$ is polynomial w.r.t. the size of $\mathcal{G}$, *i.e.*

$$\begin{aligned} |\mathcal{F}| &\leq \sum_{i=0}^{l} |\mathsf{Range}(\Omega_i)| \\ &\leq l \cdot 2|Q| \\ &\leq |Q|^2 \cdot 2|Q| = 2|Q|^3. \end{aligned}$$

[7]Since δ_i is deterministic, we sometimes write $\delta_i(p, a)$ to denote the unique $q \in Q_i$ such that $(p, a, q) \in \delta_i$.

From [PP06] we have that Streett games can be solved in time $\mathcal{O}(mn^{k+1}kk!)$ where n is the number of states, m the number of transitions and k the number of pairs in $\mathcal{F}$. Thus, in this case we have that $\Gamma_{\mathcal{G}}$ can be solved in

$$\mathcal{O}\left((2|Q|^{|Q|}|Q|!)^{3+2|Q|^3} \cdot 2|Q|^3 \cdot (2|Q|^3)!\right).$$

which is still exponential time w.r.t. the size of $\mathcal{G}$. □

Memory requirements for Eve and Adam. It is known that positional strategies suffice for Eve in parity games. On the other hand, for Streett games she might require exponential memory (see, *e.g.* [DJW97]). This exponential blow-up, however, is only on the number of pairs—which we have already argued remains polynomial w.r.t. the original automaton. It follows that:

Corollary 8. *For payoff functions* Sup, Inf, LimSup, LimInf, *for all weighted automata* $\mathcal{A}$, *there exists* m *which is* $2^{\mathcal{O}(|\mathcal{A}|)}$ *such that:*

$$\mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{W}_\forall}(\mathcal{A}) = \mathbf{Reg}_{\mathfrak{S}_\exists^m,\mathfrak{W}_\forall}(\mathcal{A}).$$

■

7.4 Upper Bounds for Discounted Sum

We turn our attention, in this section, to the discounted-sum payoff function. As discounted-sum automata are not determinizable in general, and partial-observation discounted-sum games are now known to be undecidable, we cannot follow the same ideas as for prefix-independent functions. Instead, we consider several particular cases and solve the problem for those. First, we show that the regret threshold problem can be solved whenever the discounted sum automata associated to the game structure can be made deterministic. Second, we show how to solve an ε*-gap promise* variant of the regret threshold problem.

7.4.1 Deciding r-regret: determinizable cases

When the weighted automaton $\mathcal{G}$ associated to the game structure can be made deterministic, we can solve the regret threshold problem with the following algorithm. In Section 7.2 we established that, against eloquent adversaries, computing the regret reduced to computing the value of a quantitative simulation game as defined in [CDH10]. The game is obtained by taking the product of the original automaton and a deterministic version of it. The new weight function is the difference of the weights of both components (for each pair of transitions). In [BH14], it is shown how to determinize discounted-sum automata when the discount factor is of the form $\frac{1}{n}$, for $n \in \mathbb{N}$. So, for this class of discount factor, we can state the following theorem:

Theorem 7.7. *Deciding if the regret value is less than a given threshold (strictly or non-strictly), playing against an eloquent adversary, is in* EXPTIME *for* λ *of the form* $\frac{1}{n}$ *even if it is part of the input.* ■

The complexity follows from the state complexity of the determinization procedure from [BH14].

7.4.2 The ε-gap promise problem

Given a discounted-sum automaton $\mathcal{A}$, $r \in \mathbb{Q}$, and $\varepsilon > 0$, the ε-gap promise problem adds to the regret threshold problem the hypothesis that $\mathcal{A}$ will either have regret $\leq r$ or $> r + \varepsilon$. We observe that an algorithm that satisfies that:

- a YES answer implies that $\mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{W}_\forall}(\mathcal{A}) \leq r + \varepsilon$,
- whereas a NO answer implies $\mathbf{Reg}_{\mathfrak{S}_\exists,\mathfrak{W}_\forall}(\mathcal{A}) > r$.

will decide the ε-gap promise problem.

In [BH14], it is shown that there are discounted-sum automata which define functions that cannot be realized with deterministic-sum automata. Nevertheless, it is also shown in that paper that, given a discounted-sum automaton, it is always possible to construct a deterministic one that is ε-close in the following formal sense. A *discounted-sum automaton $\mathcal{A}$ is ε-close to another discounted sum automaton $\mathcal{B}$*, if for all words x the absolute value of the difference between the values assign by $\mathcal{A}$ and $\mathcal{B}$ to x is at most ε. So, it should be clear that we can apply the algorithm underlying Theorem 7.6 to $\mathcal{G}$ and a determinized version $\mathcal{D}_{\mathcal{G}}$ of it (which is ε-close to $\mathcal{G}$) and solve the ε-gap promise problem. We can then prove the following result.

Theorem 7.8. *Deciding the ε-gap regret threshold problem is in* EXPTIME *and in in* PSPACE *if λ is not part of the input.* ■

The complexity of the algorithm follows from the fact that the value of a (quantitative simulation) game, played on the product of $\mathcal{G}$ and $\mathcal{D}_{\mathcal{G}}$ we described above, can be determined by simulating the game for a polynomial number of turns. Thus, although the automaton constructed using the techniques of Boker and Henzinger [BH14] is of size exponential, we can construct it "on-the-fly" for the required number of steps and then stop.

Proof of Theorem 7.8. We reduce the problem to determining the winner of a reachability game on an exponentially larger arena. Although the arena is exponentially larger, all paths are only polynomial in length, so the winner can be determined in alternating polynomial time, or equivalently, polynomial space.

The idea of the construction is as follows. Given a discounted-sum automaton $\mathcal{A}$, we determinize its transitions via a subset construction, to obtain a deterministic, multi-valued discounted-sum automaton $\mathcal{D}_{\mathcal{A}}$. Then we decide if Eve is able to simulate, within the regret bound, the $\mathcal{D}_{\mathcal{A}}$ on $\mathcal{A}$ for all *finite* words up to a length (polynomially) dependent on ε. If we simulate the automaton for a sufficient number of steps, then any significant gap between the automata will be unrecoverable regardless of future inputs, and we can give a satisfactory answer for the ε-gap regret problem.

More formally, given a discounted-sum automaton $\mathcal{A} = (Q, q_0, A, \delta, w)$, a regret value r and a precision $\varepsilon > 0$, we construct a reachability game $\mathcal{G}^{\varepsilon}_{\mathcal{A}}(r)$ as follows. Let

$$N := \left\lfloor \log_\lambda \left(\frac{\varepsilon(1-\lambda)}{4 w_{\max}} \right) \right\rfloor + 1,$$

where $w_{\max}$ is the maximum absolute value weight occurring in $\mathcal{A}$, so that $\frac{\lambda^N \cdot w_{\max}}{1-\lambda} < \frac{\varepsilon}{4}$. Let $P = \{\mathsf{DS}_\lambda(\pi) \mid \pi \in Q^* \text{ is a finite run of } \mathcal{A} \text{ with } |\pi| \leq N\}$

denote the (finite) set of possible discounted payoffs of words of length at most N. Let $\mathcal{F}$ be the set of functions $f : Q \to \mathbb{R} \cup \{\bot\}$, and for $f \in \mathcal{F}$, let $\mathsf{supp}(f) = \{q \in Q \mid f(q) \neq \bot\}$. Intuitively, each $f \in \mathcal{F}$ represents a weighted subset of Q ($\mathsf{supp}(f)$ being the corresponding unweighted subset), where $f(q)$ for $q \in \mathsf{supp}(f)$ corresponds to the maximal weight over all (consistent) paths ending in q (scaled by a power of λ). Given $f \in \mathcal{F}$ and $\alpha \in A$ the α-successor of f is the function f_α defined as:

$$f_\alpha(q') := \begin{cases} \max\limits_{\substack{q \in \mathsf{supp}(f) \\ (q,\alpha,q') \in \delta}} \{\lambda^{-1} \cdot f(q) + w(q, \alpha, q')\} & \text{if this set is not empty} \\ \bot & \text{otherwise.} \end{cases}$$

We define $\mathcal{F}_0 = \{f_0\}$ where $f_0(q_0) = 0$ and $f_0(q) = \bot$ for all $q \neq q_0$; and for all $n \geq 0$, we define $\mathcal{F}_{n+1} := \{f_\alpha \mid f \in \mathcal{F} \text{ and } \alpha \in A\}$. For convenience, let $F = \biguplus_{i=0}^{N} \mathcal{F}_i$ (*i.e.* a disjoint union).

The game $\mathcal{G}^\varepsilon_\mathcal{A}(r) = (V, V_\exists, E, v_0, T)$ is defined as follows:

- $V = (Q \times F \times P) \cup (Q \times F \times P \times A)$;
- $V_\exists = (Q \times F \times P \times A)$;
- $\big((q, f, c), (q, f, c, \alpha)\big) \in E$ for all $q \in Q$, $f \in F \setminus \mathcal{F}_N$, $c \in P$, and $\alpha \in A$;
- $\big((q, f, c, \alpha), (q', f', c')\big) \in E$ for all $q, q' \in Q$, $f \in F \setminus \mathcal{F}_N$, $c \in P$, and $\alpha \in A$ such that $(q, \alpha, q') \in \delta$, $f' = f_\alpha$, and $c' = c + \lambda \cdot w(q, \alpha, q')$;
- $v_0 = (q_0, f_0, 0)$; and
- $(q, f, c) \in T$ if, and only if, $f \in \mathcal{F}_N$ and $\max_{s \in \mathsf{supp}(f)} \lambda^{N-1} \cdot f(s) \leq c + r + \frac{\varepsilon}{2}$.

We claim that determining the winner of $\mathcal{G}^\varepsilon_\mathcal{A}(r)$ yields a correct response for the ε-gap promise problem.

Claim 14. *Let $\mathcal{G}^\varepsilon_\mathcal{A}(r)$ be defined as above. Then:*

- *If Eve wins $\mathcal{G}^\varepsilon_\mathcal{A}(r)$ then $\mathbf{Reg}_{\mathfrak{S}_\exists, \mathfrak{W}_\forall}(\mathcal{A}) \leq r + \varepsilon$, and*
- *if Adam wins $\mathcal{G}^\varepsilon_\mathcal{A}(r)$ then $\mathbf{Reg}_{\mathfrak{S}_\exists, \mathfrak{W}_\forall}(\mathcal{A}) > r$.*

■

Proof of Claim 14. It is easy to see that a play of $\mathcal{G}^\varepsilon_\mathcal{A}(r)$ results in Adam choosing a word $w \in A^*$ of length N, and Eve selecting a run, π, of w on $\mathcal{A}$ by resolving non-determinism at each symbol. Further, if the play terminates at (q, f, c) then $c = \mathsf{DS}_\lambda(\pi)$ and, as f contains the maximal weights of all paths (scaled by a power of λ), $\mathcal{A}(w) = \lambda^{N-1}(\max_{s \in \mathsf{supp}(f)} f(s))$. Since $|w| = N$ we have, for any infinite word $w' \in A^\omega$ and for any run, π', of $\mathcal{A}$ on w' from q, π':

$$\begin{aligned} |\mathcal{A}(w \cdot w') - \mathcal{A}(w)| &\leq \frac{\lambda^N \cdot w_{\max}}{1 - \lambda} < \frac{\varepsilon}{4}, \text{ and} \\ |\mathsf{DS}_\lambda(\pi \cdot \pi') - \mathsf{DS}_\lambda(\pi)| &\leq \frac{\lambda^N \cdot w_{\max}}{1 - \lambda} < \frac{\varepsilon}{4}. \end{aligned}$$

It follows that:

$$(\mathcal{A}(w) - \mathsf{DS}_\lambda(\pi)) - \frac{\varepsilon}{2} < \mathcal{A}(w \cdot w') - \mathsf{DS}_\lambda(\pi \cdot \pi') < (\mathcal{A}(w) - \mathsf{DS}_\lambda(\pi)) + \frac{\varepsilon}{2}. \quad (7.3)$$

Now suppose Eve wins $\mathcal{G}^\varepsilon_\mathcal{A}(r)$. Then, for every word w with $|w| = N$, Eve has a strategy σ that construct a run, π, on $\mathcal{A}$ such that $\mathcal{A}(w) \leq \mathsf{DS}_\lambda(\pi) + r + \frac{\varepsilon}{2}$. We extend this strategy to infinite words by playing arbitrarily after the first N symbols. It follows from Equation 7.3 that for every infinite word $\hat{w}$, the resulting run, $\hat{\pi}$,

$$\mathcal{A}(\hat{w}) - \mathsf{DS}_\lambda(\hat{\pi}) < (\mathcal{A}(w) - \mathsf{DS}_\lambda(\pi)) + \frac{\varepsilon}{2} \leq r + \varepsilon.$$

Since $\mathbf{reg}^\sigma_{\mathfrak{S}_\exists, \mathfrak{W}_\forall}(\mathcal{A}) = \sup_{\hat{w} \in A^\omega}(\mathcal{A}(\hat{w}) - \mathsf{DS}_\lambda(\pi))$, we have $\mathbf{Reg}_{\mathfrak{S}_\exists, \mathfrak{W}_\forall}(\mathcal{A}) \leq r + \varepsilon$.

Conversely, suppose Adam wins $\mathcal{G}^\varepsilon_\mathcal{A}(r)$. Then for any strategy of Eve, Adam can construct a word w, with $|w| = N$ such that the run, π, of $\mathcal{A}$ on w determined by Eve's strategy satisfies $\mathcal{A}(w) > \mathsf{DS}_\lambda(\pi) + r + \frac{\varepsilon}{2}$. Again, from Equation 7.3 it follows that for any infinite word $\hat{w}$ with w as its prefix and any consistent run π',

$$\mathcal{A}(\hat{w}) - \mathsf{DS}_\lambda(\hat{\pi}) > (\mathcal{A}(w) - \mathsf{DS}_\lambda(\pi)) - \frac{\varepsilon}{2} > r.$$

As this is valid for any strategy of Eve, we have $\mathbf{Reg}_{\mathfrak{S}_\exists, \mathfrak{W}_\forall}(\mathcal{A}) > r$ as required. □

Now every path in $\mathcal{G}^\varepsilon_\mathcal{A}(r)$ has length at most N, and as the set of successors of a given state can be computed on-the-fly in polynomial time, the winner can be determined in alternating polynomial time. Hence a solution to the ε-gap promise problem is constructible in polynomial space. □

Part II

Partial Observability

Chapter 8

Background II: Partial-Observation Games are Hard

The title of this chapter succinctly and accurately summarizes what is known regarding games with partial observation. Recall that a partial-observation game is played by Eve and Adam on a weighted automaton with its states partitioned into observations (*i.e.* sets of equivalent states). To start the game, a token is placed on the initial state. Then, the game proceeds in rounds from the current state p: Eve chooses a letter a and Adam resolves non-determinism by choosing a transition (p, a, q) and moving the token to q.

In game theory the concepts of imperfect, partial and limited information indicate situations where players have asymmetric knowledge about the state of the game. In the context of quantitative games this partial knowledge is reflected in Eve being unable to determine the precise location of the token amongst several equivalent states, and such games have also been extensively studied [Rei84, KV00, BD08, BCD$^+$08, DDG$^+$10].

Adding partial observability to Boolean games greatly increases the complexity of interesting problems related to them. Indeed, already for safety games with partial observation, determining the winner is EXPTIME-complete [CD10]. For reachability and parity games the problem is also EXPTIME-complete. The general recipe which is used to obtain an algorithm for these games is as follows. First, a subset construction is applied to the game in order to obtain the *belief game*. More precisely, note that in a partial-observation game, after Eve chooses an action and Adam reveals an observation to her, it might be the case that only a strict subset of the states in the observation are reachable via the sequence of actions and observations already witnessed. Hence, keeping track of play prefixes can help Eve have a more accurate belief of what the state of the game really is. More generally, this first step can be seen as determinizing the automaton on which the game is played—this can be done for all ω-automata considered in this dissertation. This yields a new, possibly exponentially bigger, deterministic parity automaton. Second, we consider a new full-observation parity game played on the product of the belief game and the deterministic parity automaton (which essentially captures the possible ways in which Adam could have resolved non-determinism in the original automaton). One can then show that Eve has a winning observation-based strategy (which might require memory) in the original game if and only if she has a winning strategy in the newly constructed, exponentially larger, full-observation parity game [RCDH07]. It follows that for quantitative games which extend Boolean games, similar solutions exist. (See Section 7.4, where we implicitly determinize Inf, Sup, LimInf, and LimSup automata.)

In the present part of this document, we mainly focus on partial-observation mean-payoff games and the related *energy games*. Energy games are two-player quantitative games of infinite horizon played on finite weighted automata with observations. The game is played in rounds in which one player, Eve, chooses letters from the automaton's alphabet whilst Adam, the second player, resolves non-determinism. The goal of Eve in an energy game is to keep a certain resource from being depleted. More specifically, she wins if, for every turn, the sum of the weights of the transitions traversed so far plus her *initial credit* is non-negative. Adam has the opposite objective: witnessing a negative value. Energy games are useful for systems in which one is interested in the use of bounded resources such as power or fuel [CdAHS03, BFL+08]. Two decision problems for energy games have been studied by the formal verification community: the FIXED INITIAL CREDIT and UNKNOWN INITIAL CREDIT PROBLEMS. In the full-observation setting, it is known that if Eve has a winning strategy in an energy game, she also has a memoryless winning strategy. Furthermore, in order to win, Eve essentially has to ensure staying in cycles with non-negative (total) weight. Using these two facts, one can show the unknown initial credit reduces in polynomial time to the fixed initial problem. (If there is some initial credit for which Eve wins, then $nw_{\max}$ should suffice—where n is the number of states and $w_{\max}$ denotes the maximum absolute value of a transition weight in the automaton.) It is also known that the fixed initial credit problem is log-space equivalent to the threshold problem for mean-payoff games [BFL+08]. In a mean-payoff game, the objective of Eve consists in maximizing the limit (inferior) of the averages of the running sum of transition weights observed along an infinite play.

It is known that some non-deterministic mean-payoff automata cannot be transformed into a deterministic mean-payoff automaton. Hence, we cannot apply to them the same solution as for Boolean games with partial observation. The study of Degorre et al. [DDG+10] in fact revealed that determining the winner of a partial-observation mean-payoff game (MPG) is undecidable. They also showed that a problem regarding energy games with partial observation is undecidable.

These unfavourable results motivate the main investigation of the following chapters: identifying classes of MPGs with partial observation where determining the winner is decidable and where strategies with finite memory, possibly memoryless, are sufficient. Our focus will be on games at the observation level, in particular we are interested in *observation-based strategies* for both players. Whilst observation-based strategies for Eve are usual in the literature, observation-based strategies for Adam had not been considered. Such strategies are more advantageous for Adam as they encompass several simultaneous concrete strategies.

We will now show that although MPGs with partial observation are not determined under the usual definition of strategy, they are determined when

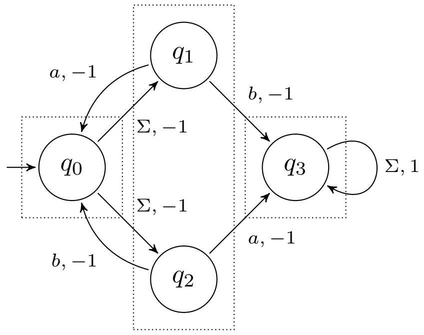

Figure 8.1: A non-determined MPG with partial observation ($\Sigma = \{a, b\}$).

Adam can use an observation-based strategy.

8.1 Observable Determinacy

One of the key features of MPGs with full observation is that they are determined, that is, it is always the case that one player has a winning strategy. This is not true in games of partial observation as can be seen in Figure 8.1. Any strategy of Adam reveals to Eve the successor of q_0 and she can use this information to play to q_3. Conversely Adam can defeat any strategy of Eve by playing to whichever of q_1 or q_2 means the play returns to q_0 on Eve's next choice (recall Eve cannot distinguish q_1 and q_2 and must therefore choose an action to apply to the observation $\{q_1, q_2\}$). This strategy of Adam can be encoded as an *observation-based strategy for Adam*: "from $\{q_1, q_2\}$ with action a or b, play to $\{q_0\}$".

Definition (Observation-based strategies for Adam)**.** Consider an MPG with partial observation $\mathcal{G} = (Q, q_0, \Sigma, \Delta, w, \mathsf{Obs})$. An *observation-based strategy for Adam* is a function $\lambda_\forall : \mathsf{obs}(\mathsf{Prefs}(\mathcal{G})) \times \Sigma \to \mathsf{Obs}$ such that for any abstract prefix $\psi = o_0\sigma_0 \ldots o_n \in \mathsf{obs}(\mathsf{Prefs}(G))$ and action $\sigma \in \Sigma$, if $\lambda_\forall(\mathsf{obs}(\pi), \sigma) = o$ then $\mathsf{obs}^{-1}(\psi \cdot \sigma o)$ ■

An abstract play $\psi = o_0\sigma_0 o_1\sigma_1 \ldots$ is consistent with $\lambda_\forall$ if $\lambda_\forall(\psi[..i], \sigma_i) = o_{i+1}$ for all $i \geq 0$.

It transpires that, under an assumption about large cardinals[1], any such counter-play by Adam is always encodable as an observation-based strategy.

Theorem 8.1 (Observable determinacy)**.** *Assuming the existence of a measurable cardinal, one player always has a winning observation-based strategy in an MPG with partial observation.* ■

To prove this we recall the definition of Suslin sets. For a detailed description of both the Borel and Suslin hierarchies we refer the reader to [Kec95].

Definition (Projective hierarchy)**.** The first level of the *Projective hierarchy* consists of Σ^1_1 (Suslin) sets, which are those whose preimage is a Borel set, *i.e.*

[1]This assumption is independent of the theory of ZFC.

all sets that can be defined as a projection of a Borel set, and Π^1_1 (co-Suslin) sets: those sets whose complement is the image of a Borel set. ■

The existence of a measurable cardinal implies Σ^1_1-Determinacy [MS88]—a weak form of the *Axiom of Determinacy.* This in turn implies games with Suslin or co-Suslin winning condition sets are determined [Kec95]. The observable determinacy of MPGs with partial observation then follows from the following result:

Lemma 8.1. *The set of plays that are winning for Eve in an MPG with partial observation is co-Suslin.* ■

Proof. Let G be a MPG with partial observation and W' be the set of all concrete plays in G for which Eve wins, namely $W' = \{\pi \in \mathsf{Plays}(G) \mid MP(\pi) \geq 0\}$, and $\overline{W'}$ its complement. Note that $W' \subseteq \mathsf{Plays}(G)$ is the payoff-set defined by the MPG winning condition ($\underline{\mathsf{MP}} \geq 0$) and is therefore in the class Π^0_3 of Borel sets [Cha07]. Let W be the set of all abstract plays such that Eve wins. Formally, we have $W = \{\psi \in \mathsf{obs}(\mathsf{Plays}(G)) \mid \forall \pi \in \mathsf{obs}^{-1}(\psi) : \pi \in W'\}$.

To show that W is in Π^1_1, we adapt the proof from [PP04] (to prove that an infinite tree is co-Suslin) and consider the set

$$U = \{(\psi, \pi) \in \mathsf{obs}(\mathsf{Plays}(G)) \times \overline{W'} \mid \exists \pi' \in \mathsf{obs}^{-1}(\psi) : \pi' = \pi\}$$

which is in the class Π^0_2 of the Borel hierarchy. To demonstrate this, let U_n be the set of pairs (ψ, π) satisfying the following property: there exists $\pi' \in \mathsf{obs}^{-1}(\psi)$ such that $\pi'[..n] = \pi[..n]$, where $n \in \mathbb{N}$. Then $U = \bigcap_{n \geq 0} U_n$, which proves U is in the class Π^0_2 since the sets U_n are open (Σ^0_1). Finally, we observe the projection of U on its first component is the complement of W, which is thus co-Suslin. □

Unfortunately, we do not know if the assumption of such an axiom is necessary for observable determinacy to hold. Therefore, in the following chapters we do not make direct use of it.

8.2 Contributions

The next three chapters summarize our work on energy games and mean-payoff games with partial observation. We begin by presenting what is known about energy games and by providing a tight complexity bound for the fixed initial credit problem for energy games. Next, we recall the result from [DDG+10] which shows that some MPG variations are undecidable and complement their result by further showing that undecidability holds for all cases. In the same chapter, we shall consider decidable classes of MPGs. Finally, we close this part of this work by considering approximations of the mean-payoff objective: window mean-payoff objectives.

The first chapter in what follows is based on our submission to *Information Processing Letters* of a short article with our result on the complexity of the fixed initial credit problem for energy games with partial observation. The second chapter summarizes our article presented at the 2014 *Reachability Problems* workshop. An extended version of the latter was submitted to *Theoretical Computer Science*. The last chapter is based on our paper presented at the

2015 *International Symposium on Automated Technology for Verification and Analysis* which has been extended and submitted to *Acta Informatica*.

Chapter 9

Partial-Observation Energy Games

In this chapter we study partial-observation games with an energy objective. Such games are played on a weighted automaton by Eve, choosing actions, and Adam, choosing a transition labelled with the given action. Eve attempts to maintain the sum of the weights (of the transitions taken) non-negative while Adam tries to do the opposite. Recall that in partial-observation games Eve does not know the exact state of the game, she is only given an equivalence class of states which contains it. In contrast, Adam has full observation.

Two decision problems for energy games have been studied by the formal verification community: the *fixed initial credit* and *unknown initial credit problems.* The former asks whether, given a fixed initial credit for Eve, she has a strategy which ensures all plays consistent with it are winning. The latter is more ambitious in that it asks whether there exists some initial credit for which the same question has a positive answer. Presently, we will first present the proof of undecidability for the unknown initial credit problem taken from [DDG$^+$10]. We will then show the fixed initial credit problem for these games is ACK-complete.

Contributions. Section 9.2 is based on [DDG$^+$10]. However, the proof of ACK-completeness for the fixed initial credit problem, presented in Section 9.3 is novel and has been submitted to the journal *Information Processing Letters.*

9.1 The Energy Objective

Though energy games are quantitative games, the classical definition for them does not use a payoff function. Instead, a direct definition for the objective—parameterized by an initial credit—is usually given. For uniformity, we follow this convention as well.

The *energy level* of a play prefix $\pi = q_0\sigma_0 \dots q_n$ is

$$\mathsf{EL}(\pi) = \sum_{i=0}^{n-1} w(q_i, \sigma_i, q_{i+1}).$$

The *energy objective* is parameterized by an initial credit $c_0 \in \mathbb{N}$ and is defined as:

$$\mathsf{PosEn}(c_0) := \{\pi \in \mathsf{Plays}(\mathcal{G}) \mid \forall i > 0 : c_0 + \mathsf{EL}(\pi[0..i]) \geq 0\}.$$

In other words, the energy objective asks for the energy level of a play never to drop below 0 when starting with energy level c_0.

9.2 Undecidability of the Unknown Initial Credit Problem

In this section we will argue that the unknown initial credit problem for partial-observation energy games is undecidable. Let us start by formalizing the problem we study.

Problem (Unknown initial credit problem)**.** Given a game $\mathcal{G}$, decide whether there exists an initial credit c_0 such that there exists a winning observation-based strategy for Eve for the objective $\mathsf{PosEn}(c_0)$. ■

To show the above problem is undecidable, we reduce from the halting problem for 2CMs. We construct, from a given 2CM, a blind energy game in which Eve has an observation-based winning strategy if and only if the machine halts. For the rest of this section, let us consider a fixed 2CM $\mathcal{M} = (Q, q_I, q_F, C, \delta)$.

Proposition 9.1 (From [DDG+10])**.** *The unknown initial credit problem is undecidable, even for blind games.* ■

Proof. Follows from the construction described below and Proposition 9.2. □

Energy game simulating the halting problem. We will now construct a blind energy game $\mathcal{G}_{\mathcal{M}}$ such that $\mathcal{M}$ has a halting run if and only if there is an initial credit for which there exists a winning observation-based strategy for Eve in $\mathcal{G}_{\mathcal{M}}$. Her winning observation-based strategy will be to perpetually simulate the halting run of the machine while the initial credit will be the length of the halting run. We set the alphabet Σ of $\mathcal{G}_{\mathcal{M}}$ to be the set of transitions of $\mathcal{M}$ plus a fresh symbol #, that is $\Sigma = \delta \cup \{\#\}$. The game $\mathcal{G}_{\mathcal{M}}$ starts with a non-deterministic transition into one of several gadgets we will describe now. The weight of the transition into each gadget is shown (labelling the initial arrow) in the Figures for the gadgets. Each gadget will check the sequence of letters played by Eve has some specific property, lest some play will get a negative energy level. Since the game is blind, Eve will not know which gadget has been chosen and will therefore have to make sure her strategy (infinite word) has all the properties checked by each gadget.

Forcing Eve to simulate δ: gadget 9.1 & 9.2. The gadget depicted in Figure 9.1 makes sure that Eve plays # as her first letter. Indeed, if she plays a strategy which does not comply then there is a play which will end up in the -1 loop and thus get a negative energy level. If she plays #, all plays in this gadget go to the 0 loop and will never have a negative energy level. To make sure that after # Eve plays the first transition from $\mathcal{M}$, and then the second, ..., we have $|\delta| + 1$ instances of the gadget from Figure 9.2. (Additionally, after having

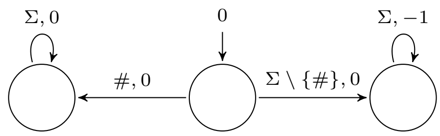

Figure 9.1: Gadget which ensures the first letter played by Eve is #.

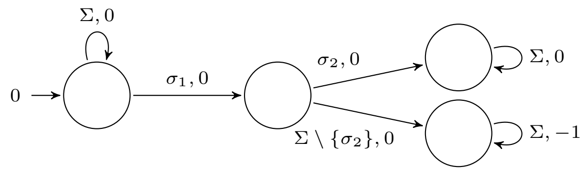

Figure 9.2: Gadget which ensures σ_1 is followed by σ_2.

played a transition leading to q_F in $\mathcal{M}$—*i.e.* of the form $(\cdot, \cdot, \cdot, q_F)$—she must play # once more.) The intuition behind the gadget is simple: if she violates the order of the sequence of transitions then there will be a play consistent with her strategy which, in the corresponding gadget, reaches the -1 loop. If she plays the letters in the correct sequence then all plays in all gadgets can only stay in the initial state or go to the upper 0 loop.

Forcing Eve to perpetually restart the simulation: gadget 9.3. To verify that Eve plays the letter # infinitely often, symbolizing the start of a new simulation of $\mathcal{M}$ every time, the game $\mathcal{G}_{\mathcal{M}}$ includes the gadget shown in Figure 9.3. If Eve plays the letter # infinitely often and within $c_0 + |\mathcal{M}|^3$ turns of each other, all the plays in the gadget will take the only negatively-weighted transition in the gadget at most $c_0 + |\mathcal{M}|^3$ times. Hence, all plays in this gadget will never have a negative energy level. If, however, Eve plays in any other way (eventually stopping with the letter # or taking too long to produce it) then there will be a play which reaches the middle state of the gadget and takes the negative transition enough times to get a negative energy level.

Forcing Eve to correctly simulate zero checks: gadget 9.4. Finally, we make sure that Eve does not play a transition which executes a zero check correctly. To do so, for each $k \in C$ we add to $\mathcal{G}_{\mathcal{M}}$ an instance of the gadget from Figure 9.4. The intuition of how the gadget works is as follows. If Eve

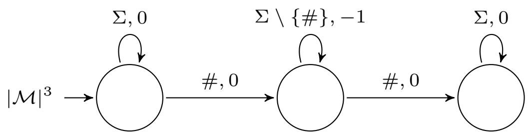

Figure 9.3: Gadget which ensures that Eve restarts her simulation of $\mathcal{M}$ infinitely often and that each simulation is of length at most $c_0 + |\mathcal{M}|^3$.

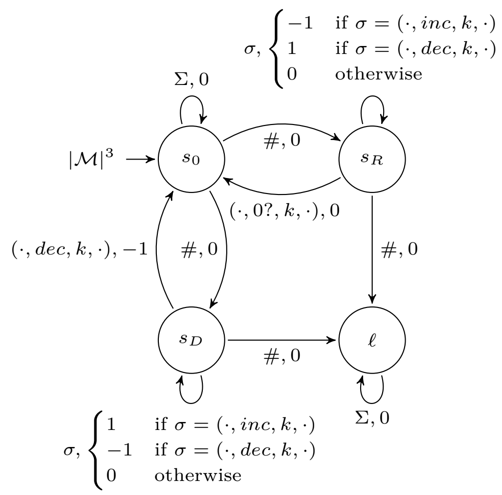

Figure 9.4: Gadget which ensures Eve correctly resolves the guarded decreases: executing a dec_k instruction only when $k > 0$ and executing the $0?_k$ instruction otherwise.

executes a $0?_k$ instruction when the counter value is positive (a *zero cheat*), then there will be a play which goes to s_R in the gadget, then simulates the inverse of the intended operations on k and moves back to the initial state on the zero cheat. This play thus far has energy level $c_0 + |\mathcal{M}|^3 - 1$. If Eve executes a dec_k instruction when the counter value is 0 (a *positive cheat*), then there will be a play which goes to s_D in the gadget, then simulates the operations on k and moves back to the initial state on the positive cheat. The latter play has energy level $c_0 + |\mathcal{M}|^3 - 1$ at this point as well. It follows that if Eve cheats more than $c_0 + |\mathcal{M}|^3$ times, there will be a play in the gadget with negative energy level. If, however, she correctly simulates the zero checks, then a play can forever stay in s_0—in which case it will never have a negative energy level—or it can move to s_R or s_D at some point. There, if Eve is simulating a finite halting run ϱ, then she will play # again in at most $|\varrho| + 1$ steps. If the play is still in s_R or s_D at that moment, then we know it moves to ℓ and that it must have seen a weight of -1 at most $|\varrho|$ times. Clearly, once in ℓ the play can no longer have a negative energy level. If the play returned to s_0 before that, then since Eve is not cheating, the energy level of the play must have been at least $c_0 + |\mathcal{M}|^3$ and it must have seen a weight of -1 at most $|\varrho|$ times.

Proposition 9.2 (From [DDG$^+$10])**.** *$\mathcal{M}$ has a halting run if and only if there exists $c_0 \in \mathbb{N}$ such that there is a winning observation-based strategy for Eve in $\mathcal{G}_{\mathcal{M}}$ given initial credit c_0.* ■

Proof. If $\mathcal{M}$ does have a halting run ϱ, then by definition ϱ has finite length n. Eve can then play the blind strategy which corresponds to the infinite word $(\#\varrho)^\omega$. Observe that this word satisfies all the constraints ensured by gadgets 9.1 and 9.2. Furthermore, there is an initial credit, namely $|\varrho|^3$, such that the word satisfies the constraints imposed by gadgets 9.3 and 9.4. Thus, no play will ever have negative energy level.

Suppose $\mathcal{M}$ has no halting run. Eve cannot play a strategy which does not correspond to a valid run of $\mathcal{M}$ or there will be a play with negative energy level in gadgets 9.1, 9.2, or 9.4. Thus, let us assume she does simulate $|\mathcal{M}|$ faithfully. For any initial credit c_0, since her simulation of $|\mathcal{M}|$ takes longer than $c_0 + |\mathcal{M}|^3$ steps (because it is not halting), the strategy will not be winning because of gadget 9.3. Hence, she has no observation-based winning strategy. □

9.3 The Fixed Initial Credit Problem

We now turn our attention to the fixed initial credit problem for energy games.

Problem (Fixed initial credit problem)**.** Given a game $\mathcal{G}$ and an initial credit c_0, decide whether there exists a winning observation-based strategy for Eve for the objective $\mathsf{PosEn}(c_0)$. ■

Decidability of the above problem was established in [DDG+10] by describing a reduction to finite safety games. We will now show that the size of the safety game used in the algorithm from [DDG+10] is at most Ackermannian with respect to the size of the input game. We then describe how the Minsky machine simulation used to show undecidability of the unknown initial credit problem can be modified to show ACK-hardness of the fixed initial credit problem. This establishes ACK-completeness of the problem.

9.3.1 Upper bound

The fixed initial credit problem was shown to be decidable in [DDG+10]. To do so, the problem is reduced to determining the winner of a safety game played on a finite tree whose nodes are functions which encode the *belief* of Eve in the original game. The notion of belief corresponds to the information Eve has about the current state (at any turn) of a partial-observation game. In this particular case, the belief of Eve is defined for any prefix $\pi = q_0\sigma_0 \dots \sigma_{n-1}q_n$ with $o = \mathsf{obs}(q_n)$. It corresponds to the subset of states $s \subseteq o$ which are reachable from q_0 via a prefix π' with the same observation sequence as π, i.e. $\mathsf{obs}(\pi) = \mathsf{obs}(\pi')$, together with the energy levels of all (worst-case) prefixes ending in q for all $q \in s$. Note that Eve only really cares about the minimal energy levels of prefixes ending in states from s. This information can thus be encoded into functions.

In this section we will first formalize the construction described above. We will then give an alternative argument (to the one presented in [DDG+10]) which proves that the constructed tree is finite. The latter goes via a translation from functions to vectors and prepares the reader for the next result. Finally, we will define an Ackermannian function and show that the size of the tree is at most the value of the function on the size of the input game.

Throughout the section we consider a fixed partial-observation energy game $\mathcal{G} = (Q, q_0, \Sigma, \Delta, w, \mathsf{Obs})$ and fixed initial credit $c_0 \in \mathbb{N}$.

Reduction to safety game

Belief functions. We define the set of *belief functions* of Eve as $\mathcal{F} := \{f : Q \to \mathbb{Z} \cup \{\bot\}\}$. The *support* of a function $f \in \mathcal{F}$ is the set $\{q \in Q \mid f(q) \neq \bot\}$.

A function $f \in \mathcal{F}$ is said to be negative if $f(q) < 0$ for some $q \in \mathsf{supp}(f)$. The initial belief function f_0 has support $\{q_0\}$ and $f_0(q_0) = c_0$. Given two functions $f, g \in \mathcal{F}$ we define the order $f \preceq g$ to hold if $\mathsf{supp}(f) = \mathsf{supp}(g)$ and $f(q) \leq g(q)$ for all $q \in \mathsf{supp}(f)$. Additionally, for $\sigma \in \Sigma$ we say g is a *σ-successor* of f if

$$\exists o \in \mathsf{Obs} : \mathsf{supp}(g) = \mathsf{post}_\sigma(\mathsf{supp}(f)) \cap o, \text{ and}$$

$$g(q) = \min\{f(p) + w(p, \sigma, q) \mid p \in \mathsf{supp}(f) \wedge (p, \sigma, q) \in \Delta\} \text{ for all } q \in \mathsf{supp}(g).$$

Intuitively, if Eve has belief function f and she plays σ, then if Adam reveals observation o to her as the observation of the new state of the game, she now has belief function g.

Sequences of functions and actions. For a function-action sequence $s = f_0\sigma_0 f_1 \dots \sigma_{n-1} f_n$ we will write f_s to denote f_n, i.e. the last function of the sequence. Let S be the smallest subset of $(\mathcal{F} \cdot \Sigma)^* \cdot \mathcal{F}$ containing f_0 and $s \cdot \sigma \cdot f$ if $s \in S$, f is a σ-successor of f_s, and it holds that: (a) f_s is not negative and (b) $f_{s'} \not\preceq f_s$ for all proper prefixes s' of s. The desired full-observation safety game is then $\mathcal{H} = (S, f_0, \Sigma, E, W)$ where

- the transition relation $E \subseteq S \times \Sigma \times S$ contains triples (s, σ, s') where $s' = s \cdot \sigma \cdot f_{s'}$, and
- the safe states are $W = \{s \in S \mid f_s \text{ is not negative}\}$.

In order for E to be total, we add self-loops (s, σ, s) for any $s \in S$ without outgoing transitions.

Lemma 9.1 (From [DDG+10])**.** *There is a winning observation-based strategy for Eve for the objective* $\mathsf{PosEn}(c_0)$ *in* $\mathcal{G}$ *if and only if there is a winning strategy for Eve in the safety game* $\mathcal{H}$*.* ■

Showing the safety game is finite

Henceforth, let $\mathcal{H} = (S, f_0, \Sigma, E, W)$ be the safety game constructed from the partial-observation game $\mathcal{G}$ and initial credit c_0 we have fixed for all of Section 9.3.1.

In the sequel, it will be useful to consider vectors instead of functions. We will therefore define an encoding of belief functions into vectors. Formally, let us fix two bijective mappings $\alpha : \{1, \dots, |Q|\} \to Q$ and $\beta : \mathcal{P}(Q) \to \{1, \dots, 2^{|Q|}\}$. (The latter two mappings essentially corresponding to fixing an ordering on Q and the set of subsets of Q.) For a belief function f, we will define a vector $\vec{f} \in \mathbb{Z}^{|Q|+2}$ which holds in its i-th dimension the value assigned by f to state $\alpha(i)$ in its support. For technical reasons (see Lemma 9.5), if state $\alpha(i)$ is not part of the support of f, we will use as place-holder the minimal value assigned by f to any state. Additionally, we use two dimensions to identify uniquely the support set of f. More formally, the vector $\vec{f}$ is

$$\left(2^{|Q|} - \beta\left(\mathsf{supp}(f)\right), \beta\left(\mathsf{supp}(f)\right), \gamma \circ \alpha(|Q|), \dots, \gamma \circ \alpha(1)\right)$$

where $\gamma(q)$ is $f(q)$ if $q \in \mathsf{supp}(f)$ and $\min\{f(q) \mid q \in \mathsf{supp}(f)\}$ otherwise.

It follows directly from the above definitions that two belief functions being $\preceq$-comparable is sufficient and necessary for their corresponding vectors to be $\leq$-comparable.[1] More formally,

Lemma 9.2. *For belief functions $f, g \in \mathcal{F}$ we have that $f \preceq g$ if and only if $\vec{f} \leq \vec{g}$.* ■

Using the above Lemma we can already argue that the safety game $\mathcal{H}$ is finite. Suppose, towards a contradiction, that $\mathcal{H}$ is infinite. It follows from König's Lemma that there is an infinite function-action sequence $s = f_0\sigma_0 f_1 \ldots$ such that for all $i \geq 0$ we have: (i) f_{i+1} is a σ_i-successor of f_i, (ii) f_i is not negative, and (iii) $f_j \not\preceq f_i$, for all $0 \leq j < i$. Now, let us consider the vector sequence $v = \vec{f_0}\vec{f_1}\ldots$. Note that the two dimensions used to represent the support of the function cannot be negative. Hence, together with condition (ii) above, it follows that $\vec{f_i} \in \mathbb{N}^{|Q|+2}$ for all $i \geq 0$. That is, the vectors have no negative integers. Further, from (iii) together with Lemma 9.2 it follows that there are no distinct vectors $\vec{f_i}, \vec{f_j}$ in the sequence v such that $\vec{f_i} \leq \vec{f_j}$. Thus, we get a contradiction with the fact that $\leq$ is a well-quasi-order for vectors of naturals (see Section 2.1).

Proposition 9.3 (From [DDG+10])**.** *The game $\mathcal{H}$ has finite state space.* ■

An Ackermann bound on the size of the safety game

The argument we have presented, to show the safety game is finite, carries the intuition that any function-action sequence from $\mathcal{H}$ should eventually end in a negative function or a function which is bigger than another function in the sequence (w.r.t. $\preceq$). How long can such sequences be? Using the relation between functions and vectors that we have established (and formalized in Lemma 9.2) we can apply results of Schmitz et al. [FFSS11, SS12, Sch16] which have been formulated for sequences of vectors of natural numbers. Intuitively, the bound they provide is based on "how big the jump is" from each vector in the sequence to the next. This last notion is formalized in the following definition.

Definition (Controlled vector sequence)**.** A vector sequence $\mathbf{a_0 a_1} \cdots \in \left(\mathbb{N}^d\right)^*$ is t-controlled by a unary increasing function $\kappa : \mathbb{N} \to \mathbb{N}$ if $|\mathbf{a_i}|_\infty < \kappa(t+i)$ for all $i \geq 0$.[2] ■

We will now define a hierarchy of sets of functions. Our intention is to determine at which level of this hierarchy we can find a function which can be said to control vector sequences induced by function-action sequences from $\mathcal{H}$. This will allow us to find a function—also at a specific level of the hierarchy—that bounds the length of such sequences (see Lemma 9.6).

The fast-growing functions. These can be seen as a sequence $(F_i)_{i\geq 0}$ of number-theoretic functions defined inductively below. [FW98]

$$F_0(x) := x + 1$$

$$F_{i+1}(x) := F_i^{x+1}(x) = \overbrace{F_i(F_i(\ldots F_i}^{x+1 \text{ times}}(x)\ldots))$$

[1] To be precise, $\leq$ here denotes the product ordering on vectors of integers (see Section 2.1).

[2] For a vector $\mathbf{a} = (a_d, a_{d-1}, \ldots, a_1)$, the infinity norm is the maximum value on any dimension, i.e. $\max\{a_i \mid 1 \leq i \leq d\}$.

The following Lemma summarizes some properties of the hierarchy.

Lemma 9.3 (From [FW98])**.** *For all $i \in \mathbb{N}$,*

- *for all $i \leq j$ and $0 \leq n \leq m$, $F_j(m) \geq F_i(n)$ and the latter is strict if the inequality between j and i or m and n is strict;*
- *F_i is primitive-recursive;*
- *F_i is dominated by F_{i+1}.*[3]

Furthermore, for all primitive-recursive functions f, there exists $i \in \mathbb{N}$ such that F_i dominates f. ■

We consider the following variant of the Ackermann function $F_\omega(x) := F_x(x)$. It is not hard to show that F_ω dominates all F_i—that is, for all $i \in \mathbb{N}$—and, in turn, all primitive-recursive functions.

The Grzegorczyk hierarchy. We now introduce a sequence $(\mathfrak{F}_i)_{i \geq 2}$ of sets of functions. Using the i-th fast-growing function, we define the i-th level of the hierarchy [Wai70, Sch16] as follows:

$$\mathfrak{F}_i := \bigcup_{c \in \mathbb{N}} \mathrm{FD}\textsc{time}\left(F_i^c(x)\right).$$

In other words, $\mathfrak{F}_i$ consists of all functions $\mathbb{N} \to \mathbb{N}$ which can be computed by a deterministic Turing machine in time bounded by any finite composition of the function F_i. Note that, since F_2 is of exponential growth, we could restrict space instead of time or even allow non-determinism and obtain exactly the same classes.

The following property of the classes of functions from the hierarchy will be useful.

Lemma 9.4 (From [LW70, Sch16])**.** *For all $i \geq 2$, every $f \in \mathfrak{F}_i$ is dominated by F_j if $i < j$.* ■

We will now show how to control vector sequences induced by function-action sequences from $\mathcal{H}$ (following the Karp-Miller tree analysis from [FFSS11]).

Lemma 9.5. *For all non-negative function-action sequences $s \in S$, the corresponding vector sequence from $\left(\mathbb{N}^{|Q|+2}\right)^*$ is $(c_0 + w_{\max} + |Q|)$-controlled by $k(x) := 2^x + x^2$.* ■

Proof. Let us assume that $w_{\max} > 0$. (This is no loss of generality as the energy game is trivial otherwise.) For any sequence $s = f_0\sigma_0 \dots \sigma_{n-1} f_n \in S$ we have that for all $0 \leq i \leq n$:

$$\begin{aligned}
|\vec{f_i}|_\infty &\leq \max\{2^{|Q|}, c_0 + i \cdot w_{\max}\} \\
&\leq 2^{|Q|} + (c_0 + w_{\max} + i)^2 \\
&\leq 2^{|Q| + c_0 + w_{\max} + i} + (|Q| + c_0 + w_{\max} + i)^2
\end{aligned}$$

which concludes the proof. □

[3]For two functions $f, g : \mathbb{N} \to \mathbb{N}$, we say g *dominates* f if $g(x) \geq f(x)$ for all but finitely many $x \in \mathbb{N}$.

Note that the control function from Lemma 9.5 is at the second level of the Grzegorczyk hierarchy. That is, $k \in \mathfrak{F}_2$, since F_2 is exponential. We can now apply the following tool.

Lemma 9.6 (From [FFSS11]). *For natural numbers $d, i \geq 1$, for all unary increasing functions $\kappa \in \mathfrak{F}_i$, there exists a function $L_{d,\kappa} : \mathbb{N} \to \mathbb{N} \in \mathfrak{F}_{i+d-1}$ such that $L_{d,\kappa}(t)$ is an upper bound for the length of non-increasing sequences from $\left(\mathbb{N}^d\right)^*$ that are t-controlled by κ.* ■

To conclude, we show how to bound the length of any sequence $s \in S$. By construction of $\mathcal{H}$, s is a function-action non-increasing sequence $f_0 \dots f_n$ of non-negative functions—except for the last function, which might be negative. The vector sequence $\vec{f_0} \dots \vec{f}_{n-1}$ is therefore non-increasing and, for all $0 \leq i < n$, the vector $\vec{f_i}$ has dimension $|Q| + 2$ and contains only non-negative numbers. It follows from Lemmas 9.5 and 9.6 that the length of the vector sequence is less than $h(c_0 + |Q| + w_{\max})$, where $h = L_{|Q|+2,k} \in \mathfrak{F}_{|Q|+3}$. Hence, the length of s is bounded by $h(c_0 + |Q| + w_{\max}) + 1$. Let us write $|\mathcal{G}| = |Q| + c_0 + w_{\max} + |\Delta| + |\mathsf{Obs}|$. We thus have that $h(|\mathcal{G}|) + 1$ bounds the length of all $s \in S$. Clearly then, the size of S is at most $|\Delta|^{h(|\mathcal{G}|)+1}$. More coarsely, we have that $2^{(h(|\mathcal{G}|)+1)^2}$ bounds the size of S. It follows from Lemmas 9.3–9.6 that the latter bound is primitive recursive for all fixed $\mathcal{G}$. Since F_ω dominates all primitive-recursive functions, we conclude $|S|$ is $\mathcal{O}(F_\omega(|\mathcal{G}|))$. As safety games are known to be solvable in linear time with respect to the size of the game graph (see, e.g., [AG11]), the desired result then follows from Lemma 9.1.

Theorem 9.1. *The fixed initial credit problem is decidable in Ackermannian time.* ■

9.3.2 Lower bound

In the sequel we will establish Ackermannian hardness of the fixed initial credit problem, thus giving a negative answer to the question of whether the problem has a primitive-recursive algorithm. This question is of particular interest in light of recent work by Jurdziński et al. [JLS15] in which it is shown the same problem is 2EXPTIME-complete for multi-dimensional games with full observation.

To begin, we will formally define the ACK complexity class—using the hierarchies of functions introduced in the previous section. We will then adapt the translation from Minsky machines to partial-observation energy games presented in [DDG+10] (to argue the unknown initial credit problem is undecidable) and reduce the existence of a halting run with bounded counter values in the original machine to the fixed initial credit problem in the constructed game. Finally, we will describe how to make sure the bound on the counters is Ackermannian (without explicitly computing the Ackermann function during the reduction).

The complexity class. We adopt the definition proposed by Schmitz [Sch16] for the class of Ackermannian decision problems:

$$\textsc{Ack} := \bigcup_{g \in \mathfrak{F}_{<\omega}} \textsc{Dtime}\left(F_\omega\left(g(n)\right)\right)$$

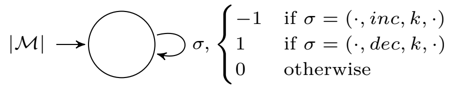

Figure 9.5: Gadget which ensures that Eve respects the bound on the counters.

where $\mathfrak{F}_{<\omega} := \bigcup_{i \in \mathbb{N}} \mathfrak{F}_i$. Note that we allow ourselves any kind of primitive-recursive reduction. It was shown in [Sch16] that: for any two functions $g, f \in \mathfrak{F}_{<\omega}$, there exists p in $\mathfrak{F}_{<\omega}$ such that $f \circ F_\omega \circ g$ is dominated by $F_\omega \circ p$. It follows the distinction between time-bounded and space-bounded computations is actually irrelevant here since F_2 is already of exponential growth.

Minsky machine simulation

We will now prove some intermediate results, regarding the bounded halting problem for Minsky machines, which will be useful to reduce the F_ω-bounded halting problem to the fixed initial credit problem. As an exercise, we will first reduce from the f-bounded problem, for f the *identity function*.

Lemma 9.7. *A 2CM $\mathcal{M}$ has an $f(|\mathcal{M}|)$-bounded halting run if and only if it has an $f(|\mathcal{M}|)$-bounded halting run of length at most $|\mathcal{M}|(f(|\mathcal{M}|))^2$.* ■

Proof. We focus on the only non-trivial direction. If no $f(|\mathcal{M}|)$-bounded run of the machine does reach q_f in at most $|\mathcal{M}|(f(|\mathcal{M}|))^2$ steps, then we have two possibilities. It could be the case that the machine has a counter whose value goes above $f(|\mathcal{M}|)$ before reaching q_F. Clearly, $\mathcal{M}$ has no $f(|\mathcal{M}|)$-bounded halting run in this case. Otherwise, the machine must have repeated at least one configuration. Since the machine is deterministic, this means it will never halt (i.e. it will cycle without reaching q_F). □

Bounding the counters. In Figure 9.5 we can see a very simple gadget which is instantiated for each $k \in C$. Its task is to ensure Eve does not play a sequence of inc_k and dec_k which results in the value of k being larger than $|\mathcal{M}| + c_0$.

Proposition 9.4. *$\mathcal{M}$ has an $|\mathcal{M}|$-bounded halting run if and only if there is a winning observation-based strategy for Eve in $\mathcal{G}_\mathcal{M}$ given initial credit $c_0 = 0$.* ■

Proof. If $\mathcal{M}$ does have an $|\mathcal{M}|$-bounded halting run ϱ, then we can assume that ϱ has length at most $|\mathcal{M}|^3$ (see Lemma 9.7). Eve can then play the blind strategy which corresponds to the infinite word $(\#\varrho)^\omega$. Since this word satisfies all the constraints ensured by the gadgets in $\mathcal{G}_\mathcal{M}$, no play will ever have negative energy level.

Suppose $\mathcal{M}$ has no $|\mathcal{M}|$-bounded halting run. Eve cannot play a strategy which does not correspond to a valid run of $\mathcal{M}$ or there will be a play with negative energy level in gadgets 9.1, 9.2, or 9.4. Thus, let us assume she does simulate $|\mathcal{M}|$ faithfully. We now consider two cases depending on whether $|\mathcal{M}|$ has a halting run (which is, necessarily, not $|\mathcal{M}|$-bounded). If $|\mathcal{M}|$ has a halting run which is not $|\mathcal{M}|$-bounded, then a play with negative energy level can be constructed in gadget 9.5. Similarly, if her simulation of $|\mathcal{M}|$ stays $|\mathcal{M}|$-bounded but takes longer than $|\mathcal{M}|^3$ steps (because it is not halting), the strategy will not be winning because of gadget 9.3. Hence, she has no observation-based winning strategy. □

We will now generalize the reduction we just presented and use it to prove the announced ACK-hardness result. In short, we need to make sure the energy level of plays entering gadget 9.5 becomes $F_\omega(|\mathcal{M}|)$ and the energy level of plays entering gadgets 9.3 and 9.4 becomes greater than $|\mathcal{M}|(F_\omega(|\mathcal{M}|))^2$. The way in which we propose to do so is to add an initial gadget to $\mathcal{G}_\mathcal{M}$ which allows Eve to play a specific sequence of letters to get the required energy level—and no more—and then forces her into the simulation of $\mathcal{M}$ which we have already described. As a first step, we describe a way of computing F_ω using vectors.

Vectorial version of the fast-growing functions

We will now give a vectorial-based definition of F_k, for any $k \in \mathbb{N}$. Intuitively, we will use $k+1$ dimensions to keep track of how many iterations of F_j (for $0 \le j \le k$) still need to be applied on the current intermediate value. More formally, for a vector $\mathbf{a} = (a_k, \ldots, a_0) \in \mathbb{N}^{k+1}$ we set

$$\Phi(\mathbf{a}; x) = \Phi(a_k, \ldots, a_0; x) := F_k^{a_k}(\ldots F_1^{a_1}(F_0^{a_0}(x))).$$

It follows that $F_\omega(x) = \Phi(1, \overbrace{0, \ldots, 0}^{x \text{ times}}; x)$. The following is the key property associated with Φ.

Lemma 9.8. *For vectors $\mathbf{a}, \mathbf{b} \in \mathbb{N}^{k+1}$ and $x, y \in \mathbb{N}$, if $\mathbf{a} \le \mathbf{b}$ and $x \le y$ then $\Phi(\mathbf{a}; x) \le \Phi(\mathbf{b}; y)$.* ■

Proof. It is a direct consequence of the definition of Φ and Lemma 9.3. □

We consider the following (family of) rewrite rules $N0$, $N1_j$, and $N2$:

$$\begin{aligned} \Phi(\mathbf{a}; x) &\to_{N0} x \\ \Phi(\ldots, a_j + 1, a_{j-1}, \ldots; x) &\to_{N1_j} \Phi(\ldots, a_j, x+1, \ldots; x) \\ \Phi(a_k, \ldots, a_0 + 1; x) &\to_{N2} \Phi(a_k, \ldots, a_0; x+1) \end{aligned}$$

For simplicity, denote the set of rewrite rules $\{N1_j \mid 0 < j \le k\}$ by $N1$. Let us write $(\mathbf{b}, y) \rightsquigarrow_r (\mathbf{a}, x)$ and $(\mathbf{b}, y) \rightsquigarrow_N (\mathbf{a}, x)$ if rule r or, respectively, a sequence of the rules $N1$ and $N2$, can be applied to $\Phi(\mathbf{b}; y)$ to transform it into $\Phi(\mathbf{a}; x)$.[4] We remark that $(\mathbf{b}, y) \rightsquigarrow_N (\mathbf{a}, x)$ implies $\mathbf{a}$ is smaller than $\mathbf{b}$ for the lexicographical order. It follows that the application of the rewrite rules always terminates.

Lemma 9.9. *For any vector $\mathbf{b} \in \mathbb{N}^{k+1}$ and $x \in \mathbb{N}$, the set $\{\mathbf{a} \in \mathbb{N}^{k+1} \mid \exists y \in \mathbb{N} : (\mathbf{b}, y) \rightsquigarrow_N (\mathbf{a}, x)\}$ is finite.* ■

Remark that rule $N1_j$ can be applied to $\Phi(\mathbf{a}; x)$ for any $0 < j \le k$ as long as $a_j > 0$. We would like to argue that the "best" way to use $N1_j$, in order to obtain the highest possible final value, is to do so only if all dimensions $0 \le \ell < j$ have value 0. Formally, let us write $(\mathbf{b}, y) \to_r (\mathbf{a}, x)$ if $(\mathbf{b}, y) \rightsquigarrow_r (\mathbf{a}, x)$ and, additionally, if $r = N1_j$ then it holds that $a_\ell = 0$ for all $0 \le \ell < j$. We

[4]Since the rule $N0$ yields a single number, it cannot be the case that $(\mathbf{b}, y) \rightsquigarrow_{N0} (\mathbf{a}, x)$.

then say the application of the rewrite rule r was *proper*. Similarly, we write $(\mathbf{b}, y) \to_N (\mathbf{a}, x)$ if $\Phi(\mathbf{a}; x)$ can be obtained by proper application of rules $N1$ and $N2$ to $\Phi(\mathbf{b}; y)$.

Lemma 9.10. *For all vectors $\mathbf{a}, \mathbf{b} \in \mathbb{N}^{k+1}$ and $x, y \in \mathbb{N}$, if $(\mathbf{b}, y) \to_N (\mathbf{a}, x)$ then $\Phi(\mathbf{b}; y) = \Phi(\mathbf{a}; x)$.* ■

Proof. Follows directly from the definitions of Φ and the fast-growing functions, and proper application of rules $N1$ and $N2$. □

The above result tells us that the proper application of the rewrite rules to $\Phi(a_k, \ldots, a_0; x)$ give us a correct computation of $F_k^{a_k}(\ldots(F_0^{a_0}(x)))$. We will now argue that improper application of the rules will result in a smaller value.

Lemma 9.11. *For all vectors $\mathbf{a}, \mathbf{b} \in \mathbb{N}^{k+1}$ and $x, y \in \mathbb{N}$, if $|\mathbf{b}|_\infty \leq x$ and $(\mathbf{b}, y) \rightsquigarrow_N (\mathbf{a}, x)$ then $\Phi(\mathbf{b}; y) \leq \Phi(\mathbf{a}; x)$.* ■

Proof. If $(\mathbf{b}, y) \to_N (\mathbf{a}, x)$ then the result follows by applying Lemma 9.10. If this is not the case, the sequence of rules applied to $\Phi(\mathbf{b}; y)$ to obtain $\Phi(\mathbf{a}; x)$ includes at least one improperly applied rule. Since $N2$ cannot be applied improperly, we focus on $N1$. We will show that applying an $N1$ rule improperly cannot increase the value. The desired result will then follow by induction.

We will argue that for any $z \in \mathbb{N}$ and any vector $\mathbf{c} \in \mathbb{N}^{k+1}$ such that $|\mathbf{c}|_\infty \leq z$, it holds that for all $2 \leq i \leq k+1$, applying any rule from $\{N1_j \mid 0 < j < i\}$ improperly to $\Phi(\mathbf{c}; z)$, yields a smaller value. For the base case we consider $i = 2$. We need to show the property holds for $N1_1$. Assume that $(\mathbf{c}, z) \rightsquigarrow_{N1_1} (c_k, \ldots, c_1 - 1, z+1, z)$ and that $c_0 > 0$. By applying c_0 times the rule $N2$ to $\Phi(\mathbf{c}; z)$ we obtain $\Phi(c_k, \ldots, c_1, 0; z + c_0)$. We can now properly apply $N1_1$ to the latter and obtain $\Phi(c_k, \ldots, c_1 - 1, z + c_0; z + c_0)$. It follows from Lemma 9.8 that the claim holds for $i = 2$. To conclude, we show that if it holds for i then it must hold for $i + 1$. We only need to show the property holds for $N1_i$. Assume that $(\mathbf{c}, z) \rightsquigarrow_{N1_i} (c_k, \ldots, c_i - 1, z+1, c_{i-2}, \ldots, z)$ and that $c_\ell > 0$ for some $0 \leq \ell < i$. By applying the sequence of rules $N1_\ell N1_{\ell-1} \ldots N1_1 N2 N1_i N1_{i-1} \ldots N1_1$ to $\Phi(\mathbf{c}; z)$ we obtain $\Phi(c_k, \ldots, c_i - 1, z+1, z+1, \ldots, z+2; z+1)$. It follows from induction hypothesis and the fact that $|\mathbf{b}|_\infty \leq z$ that $\Phi(\mathbf{c}; z)$ is larger than the latter, which is in turn larger than $\Phi(c_k, \ldots, c_i - 1, z+1, c_{i-2}, \ldots; z)$ according to Lemma 9.8. Thus, the claim holds. □

In the next section we will detail a new gadget which can be used to pump an energy level of m up to $F_m(m) = F_\omega(m)$. The gadget simulates the rewrite rules to compute F_ω vectorially.

F_ω-pumping gadget

For convenience, we will focus on the blind gadget as a blind energy game itself. We will later comment on how it fits together with the Minsky machine game constructed in Section 9.3.2.

Let us consider a fixed $m \in \mathbb{N}$. The blind energy game I_m we build has exactly $m + 5$ states, namely: an initial state q_0 and states $\{\top, f, \chi\} \cup \{\alpha_i \mid 0 \leq i \leq m\}$. The game starts with a non-deterministic choice of state from the set $\{\chi\} \cup \{\alpha_i \mid 0 \leq i \leq m\}$. The weight of the transition going to state α_m is 1; to state χ, m; to all other states, 0. The alphabet consists of as many rewrite rules

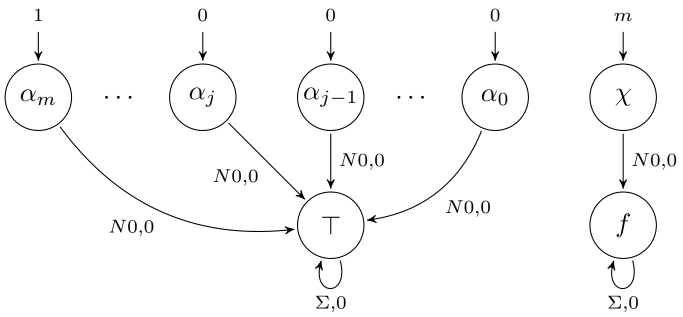

Figure 9.6: Pumping gadget with only transitions for $\sigma = N0$ shown for states $\{\chi\} \cup \{\alpha_i \mid 0 \leq i \leq m\}$.

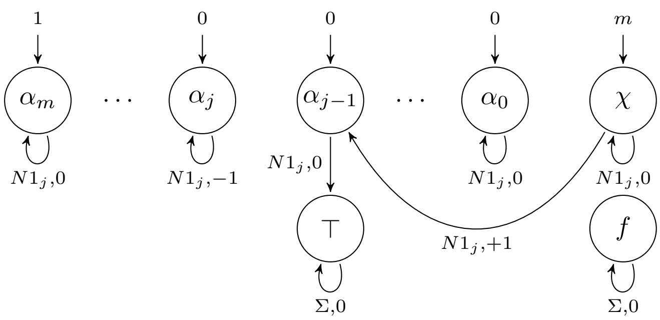

Figure 9.7: Pumping gadget with only transitions for $\sigma = N1_j$ shown for states $\{\chi\} \cup \{\alpha_i \mid 0 \leq i \leq m\}$.

(defined in the previous section) as required to compute F_m. More formally, the alphabet is $\Sigma = \{N0, N2\} \cup \{N1_j \mid 1 \leq j \leq m\}$. For clarity, the game has been split into Figures 9.6–9.8, each Figure showing transitions for different letters. Intuitively, the game I_m allows Eve to simulate the vectorial computation of $F_\omega(m)$. Once she plays the letter N_0 the play moves to either $\top$ or f, and—as we will argue later—if she has correctly simulated the computation of the function and the play has reached f, its energy level will be $F_\omega(m)$.

Let us write Σ_N for the restricted alphabet $\{N2\} \cup \{N1_j \mid 1 \leq j \leq m\} \subseteq \Sigma$ and Q_N for the set of states $\{\chi\} \cup \{\alpha_i \mid 0 \leq i \leq m\}$. We now prove the property this game (or gadget) enforces. The idea is that Eve playing a sequence of rewrite rules Σ_N has the effect that all plays consistent with her strategy and which end in α_i have energy level equal to the value of a_i after applying the rules to $\Phi(\mathbf{a}; m)$.

Lemma 9.12. *Consider any play prefix $\pi = q_0\sigma_0 \ldots \sigma_{n-1}q_n$ in I_m such that $\sigma_i \in \Sigma_N$ for all $0 < i < n$, and $\alpha_i \in Q_N$ and $\mathsf{EL}(\pi[0..i]) \geq 0$ for all $0 < i \leq n$. If $q_n = \alpha_j$ then $\mathsf{EL}(\pi) = a_j^n$, and if $q_n = \chi$ then $\mathsf{EL}(\pi) = m^n$, where $\mathbf{a^0} = (1, \overbrace{0, \ldots, 0}^{m \text{ times}})$, $m^0 = m$, and $(a_m^0, \ldots, a_0^0, m^0) \rightsquigarrow_{\sigma_0} \cdots \rightsquigarrow_{\sigma_{n-1}} (a_m^n, \ldots, a_0^n, m^n)$.* ■

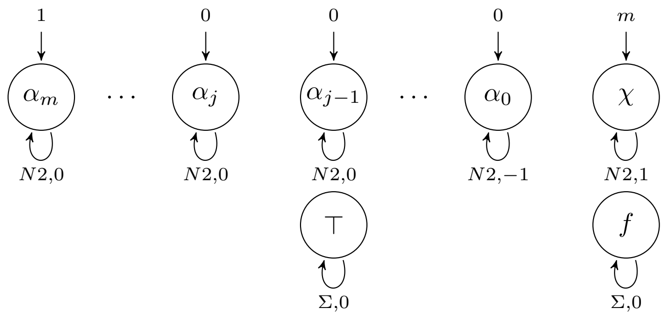

Figure 9.8: Pumping gadget with only transitions for $\sigma = N2$ shown for states $\{\chi\} \cup \{\alpha_i \mid 0 \le i \le m\}$.

Proof. We proceed by induction on n. Note that if $n = 1$, i.e. the play prefix has only two states, the energy level will be equal to 1 if the second state in the play is α_m, m if it is χ, and 0 otherwise. Hence the claim holds for prefixes of some length n. Let us argue that this also holds for prefixes of length $n+1$. Consider an arbitrary play prefix $\pi = q_0\sigma_0 \dots \sigma_n q_{n+1}$ for which all assumptions hold. By induction hypothesis we have that $\mathsf{EL}(\pi[0..n]) = a_j^n$ if $q_n = \alpha_j$ and m^n otherwise. If $\sigma_n = N2$, following the transitions shown in Figure 9.8 we get that: if $q_n = \alpha_j$, and $0 < j \le m$ then the claim holds since $q_{n+1} = \alpha_j$ and $\mathsf{EL}(\pi) = a_j^{n+1} = a_j^n$; also, if $q_n = \alpha_0$ the claim holds since $q_{n+1} = \alpha_0$ and $\mathsf{EL}(\pi) = a_0^{n+1} = a_0^n - 1$ as expected; finally, if $q_n = \chi$ then it holds since $q_{n+1} = \chi$ and $\mathsf{EL}(\pi) = m^{n+1} = m^n + 1$. Otherwise, if $\sigma_n = N1_\ell$, following the transitions shown in Figure 9.7 we get that: if $q_n = \alpha_j$, and $j \notin \{\ell, \ell - 1\}$ then the claim holds since $q_{n+1} = \alpha_j$ and $\mathsf{EL}(\pi) = a_j^{n+1} = a_j^n$; if $q_n = \chi$ then it holds since either $q_{n+1} = \chi$ and $\mathsf{EL}(\pi) = m^{n+1} = m^n$ or $q_{n+1} = \alpha_{\ell-1}$ and $\mathsf{EL}(\pi) = a_{\ell-1}^{n+1} = m^n + 1$; it cannot be the case that $q_n = \alpha_{\ell-1}$ since otherwise q_{n+1} must be $\bot$ and that would violate our assumptions; finally, if $q_n = \alpha_\ell$ then $q_{n+1} = \alpha_\ell$ and $\mathsf{EL}(\pi) = a_\ell^{n+1} = a_\ell^n - 1$ as required. The result thus follows by induction. $\square$

We are finally ready to prove our main result.

Theorem 9.2. *The fixed initial credit problem is* Ack*-hard, even for blind games.* $\blacksquare$

Proof. For any 2CM $\mathcal{M}$ we will consider two instances of the new pumping gadget I_m, with $m = |\mathcal{M}|$, and one instance of the 2CM-simulating game $\mathcal{G}_\mathcal{M}$ from Section 9.3.2. (The first copy of I_m will be used to compute the Ackermann function while the second one will be used to obtain a value greater than $m(F_\omega(m))^2$.) Additionally, we will add two new states, s_0 and s_1, which have a 0-weighted self-loop on all letters from the alphabet of I_m, except for $N0$, and a bad sink state, $\bot$, which has self-loops with weight -1 on all letters from the alphabets of I_m and $\mathcal{G}_\mathcal{M}$. The good sink $\top$, in copies of I_m now have self-loops with weight 0 on all letters from the alphabet of $\mathcal{G}_\mathcal{M}$ and I_m.

We describe how all five components are connected (see Figure 9.9). From χ in the first copy of I_m with $N0$ and weight 0 we non-deterministically go to

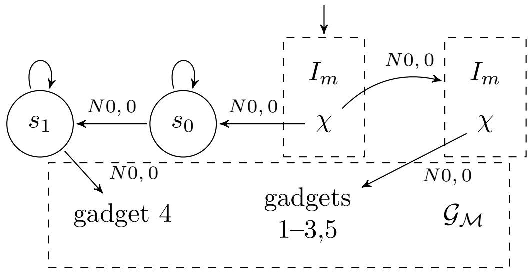

Figure 9.9: Overview of the blind game used to show ACK-hardness.

s_0 and the initial state of the second I_m; from s_0 with $N0$ and weight 0 we deterministically go to s_1; from s_1 with $N0$ and weight 0 we deterministically go to the $\mathcal{G}_\mathcal{M}$ gadget from Figure 9.5; from χ in the second copy of I_m with $N0$ and weight 0 we non-deterministically go to all copies of the gadgets from Figures 9.1, 9.2, 9.3, and 9.4. Finally, to make sure the transition relation is total, from both copies of I_m, s_0, and s_1 we add transitions to $\perp$ on all letters from the alphabet of $\mathcal{G}_\mathcal{M}$. Also, from $\mathcal{G}_\mathcal{M}$ we add transitions to $\perp$ on all letters from the alphabet of I_m.

We will now argue that Eve has an observation-based winning strategy for initial credit $c_0 = 0$ if and only if $\mathcal{M}$ has an $F_\omega(|\mathcal{M}|)$-bounded halting run.

If it halts, she wins. If $\mathcal{M}$ has an $F_\omega(|\mathcal{M}|)$-bounded halting run, then Eve should play the sequence of proper rewrite rules required to compute $m' = F_\omega(|\mathcal{M}|)$ vectorially from $\Phi(1, 0, \ldots, 0; m)$. She will then play $N0$ and choose letters to compute $F_{m'}^{m'+1}(F_{m'-1}^{m'}(\ldots F_0^{m'}(m' + m)))$ which we denote by m''. Note that $m'' \geq |\mathcal{M}|(F_\omega(|\mathcal{M}|))^2$. Finally, she will simulate the halting run of $\mathcal{M}$. From Lemma 9.12 we have that: after the first time she plays $N0$ the play is either in the first copy of $\top$ (and will never have negative energy level) or it has energy level $F_\omega(|\mathcal{M}|)$ and is now in the initial state of the second copy of I_m or s_0. By playing $N0$ the play now moves to s_1 or the simulation of the computation of m''. After playing a third $N0$, the play moves from the second copy of χ—with energy level m''—and from s_1—with m'— to the corresponding gadgets in $\mathcal{G}_\mathcal{M}$ or it moves to $\top$, where Eve cannot lose. Hence, any play consistent with this strategy of Eve, and which does not enter $\mathcal{G}_\mathcal{M}$, cannot have negative energy level. If the play has entered $\mathcal{G}_\mathcal{M}$ then the same arguments as presented to prove Proposition 9.4 should convince the reader that it cannot have negative energy level.

If it does not halt, she does not win. If $\mathcal{M}$ has no $F_\omega(|\mathcal{M}|)$-bounded halting run then, from Lemmas 9.9 and 9.12 we have that Eve eventually play three times $N0$ to exit the copies of I_m and enter $\mathcal{G}_\mathcal{M}$—or end in a $\top$ state—lest we can construct a play with negative energy level. Also, using Lemma 9.11, we conclude that she cannot exit the two copies of I_m and enter $\mathcal{G}_\mathcal{M}$ with energy level greater than $F_\omega(|\mathcal{M}|)$ for the gadget from Figure 9.5 or energy level greater than m'' for the other gadgets, respectively. It thus follows from the proof of

Proposition 9.4 that she has no observation-based winning strategy. □

Chapter 10

Partial-Observation Mean-Payoff Games

In this chapter, we identify classes of MPGs with partial observation where determining the winner is decidable and where strategies with finite memory, possibly memoryless, are sufficient.

To simplify our definitions and algorithmic results, we initially consider a restriction on the set of observations which we term *limited observation*. In games of limited observation the current observation contains only those vertices consistent with the observable history, that is the observations are the *belief set of Eve* (see, *e.g.* [CD12]). This is not too restrictive as any MPG with partial observation can be realized as a game of limited observation via a subset construction. In Section 10.4 we consider the extension of our definitions to MPGs with partial observation via this construction.

In games of full observation one aspect of MPGs that leads to simple (but not quite efficient) decision procedures is their equivalence to finite cycle-forming games. Such games are played as their infinite counterparts, however when the play (or, the token Eve and Adam move on the automaton) revisits a state, the game is stopped. The winner is determined by a finite analogue of the mean-payoff condition on the cycle now formed; that is, Eve wins if the average weight of the edges traversed in the cycle exceeds a given threshold. Ehrenfeucht and Mycielski [EM79] used this equivalence to show that positional strategies are sufficient to win MPGs with full observation. The latter leads to a $\mathrm{NP} \cap \mathrm{coNP}$ procedure for determining the winner. Critically, a winning strategy in the finite game translates directly to a winning strategy in the MPG, so such games are especially useful for strategy synthesis.

We extend this idea to games of partial observation by introducing a finite, full observation, cycle-forming game played at the observation level. That is, the game finishes when an observation is revisited (though not necessarily the first time). In this reachability game, winning strategies can be translated to finite-memory winning strategies in the MPG. This leads to a large, natural subclass of MPGs with partial observation, which we name *forcibly terminating* games, where determining the winner is decidable and finite memory observation-based strategies suffice.

Unfortunately, recognizing if an MPG is a member of this class is undecidable, and although determining the winner is decidable, we show that this problem is complete (under polynomial-time reductions) for the class of all decidable problems. Motivated by these negative algorithmic results, we investigate two natural refinements of this class for which winner determination and class membership are decidable. The first, *forcibly first abstract cycle* games (forcibly FAC games, for short), is the natural class of games obtained when our cycle-forming game is restricted to simple cycles. Unlike in full-observation mean-payoff games, we show that winning strategies in this finite simple cycle-forming game may require memory, though this memory is at most exponential in the size of the game. The second refinement, *first abstract cycle* (FAC) games, is a further structural refinement that guarantees a winner in the simple cycle-forming game. We show that in this class of games positional observation-based strategies suffice.

Preliminaries

Let make explicit the assumptions and define some notation which will be used in this chapter.

Assumptions. In this chapter, when we study the problem of deciding a game, we assume a threshold $\nu = 0$. This is no loss of generality in mean-payoff games. Indeed, given any instance of the problem in which $\nu \neq 0$, we can 'shift' and 'scale' the weight function to obtain a second game in which Eve wins with threshold 0 if and only if she wins the original game with threshold ν. Additionally, we assume that all weights are integral. Once more, this is no loss of generality since it can be achieved by 'scaling' all the weights of the original automaton.

Weight of a play prefix. For a concrete prefix $\pi = q_0 a_0 \dots q_n$ we write $w(\pi)$ for the sum of the weights of its transitions:

$$w(\pi) := \sum_{i=0}^{n-1} w(q_i, a_i, q_{i+1}).$$

Let us now formally defined the limited-observation games we have mentioned in the introduction.

Definition (Limited observation)**.** We say an MPG with partial observation $\mathcal{G} = (Q, q_I, \Sigma, \Delta, w, \mathsf{Obs})$ has *limited observation* if Obs satisfies the following:

(1) $\{q_I\} \in \mathsf{Obs}$, and

(2) For each $(o, \sigma) \in \mathsf{Obs} \times \Sigma$ the set $\{q' \mid \exists q \in o \text{ and } (q, \sigma, q') \in \Delta\}$ is a union of elements of Obs.

■

Note that the second condition is equivalent to saying that if $q \in o$, $q' \in o'$ and $(q, \sigma, q') \in \Delta$ then for every $r' \in o'$ there exists $r \in o$ such that $(r, \sigma, r') \in \Delta$.

We now show how to transform a game with partial observation into a game with limited observation. The idea behind the translation is to take subsets of the observations and restrict transitions to those that satisfy the limited-observation requirements. More formally, given an MPG with partial observation $\mathcal{G} = (Q, \Sigma, \Delta, q_I, w, \mathsf{Obs})$ we construct a second MPG with the property that it has limited observation. Formally, the limited-observation MPG is $\mathcal{G}' = (Q', \Sigma, \Delta', q'_I, w', \mathsf{Obs}')$ where:

- $Q' = \{(q, K) \in Q \times \mathcal{P}(Q) \mid q \in K \text{ and } K \subseteq o \in \mathsf{Obs}\}$,
- $q'_I = (q_I, \{q_I\})$,
- $\mathsf{Obs}' = \big\{\{(q, K) \mid q \in K\} \mid K \subseteq o \text{ for some } o \in \mathsf{Obs}\big\}$,
- Δ' contains the transitions $((q, K), \sigma, (q', K'))$ such that $(q, \sigma, q') \in \Delta$ and $K' = \mathsf{post}_\sigma(K) \cap o$ for some $o \in \mathsf{Obs}$, and
- $w'((q, K), \sigma, (q', K')) = w(q, \sigma, q')$ for all $((q, K), \sigma, (q', K')) \in \Delta'$.

It is folklore to show that this *knowledge-based* subset construction (also known as a belief construction) preserves observation-based winning strategies of Eve. The terms belief and knowledge are used to denote a state from any variation of the classic "Reif construction" [Rei84] to turn a game with partial observation into a game with full observation. Other names for similar constructions include "knowledge-based subset construction" (see *e.g.* [DDG+10]). In this case the resulting game is not one with full observation but one with limited observation.

Lemma 10.1 (Equivalence)**.** *Let $\mathcal{G}$ be an MPG with partial observation and $\mathcal{G}'$ be the corresponding MPG with limited observation as constructed above. Eve has a winning observation-based strategy in $\mathcal{G}$ if and only if she has a winning observation-based strategy in $\mathcal{G}'$.* ■

Abstract and concrete cycles. An *abstract (respectively concrete) cycle* is an abstract (concrete) path $\chi = o_0\sigma_0 \dots o_n$ where $o_0 = o_n$. We say χ is *simple* if $o_j \neq o_i$ for $0 \leq i < j < n$. Given $k \in \mathbb{N}$ define χ^k to be the abstract (concrete) cycle obtained by traversing k times χ. That is, $\chi^k = o'_0\sigma'_0 \dots o'_{nk}$ where for all $j \geq 0$, $o'_j = o_{j \pmod n}$ and $\sigma'_j = \sigma_{j \pmod n}$. A *cyclic permutation* of χ is an abstract (concrete) cycle $o'_0\sigma'_0 \dots o'_n$ such that $o'_j = o_{j+k \pmod n}$ and $\sigma'_j = \sigma_{j+k \pmod n}$ for some k. If $\chi' = o'_0\sigma'_0 \dots o'_m$ is a cycle with $o'_0 = o_i$ for some i, the *interleaving* of χ and χ' is the cycle $o_0\sigma_0 \dots o_i\sigma'_0 \dots o'_m\sigma_i \dots o_n$.

Non-zero-sum reachability games. A *(non-zero-sum) reachability game* $\mathcal{G} = (Q, q_I, \Sigma, \Delta, T_\exists, T_\forall)$ is a tuple where Q is a (not necessarily finite) set of states; Σ is a finite set of actions; $\Delta \subseteq Q \times \Sigma \times Q$ is a finitary transition function (that is, for any $q \in Q$ and $\sigma \in \Sigma$ there are finitely many q' such that $(q, \sigma, q') \in \Delta$); $q_I \in Q$ is the initial state; and $T_\exists, T_\forall \subseteq Q$ are the terminating states. The game is played as follows. We place a token on $q_I \in Q$ and start the game. Eve chooses an action $\sigma \in \Sigma$ and Adam chooses a σ-successor of the current location as determined by Δ. The process is repeated until the game reaches a state in $T_\exists$ or $T_\forall$. In the first case we declare Eve as the winner whereas the latter corresponds to Adam winning the game. Notice that the game, in general, might not terminate, in which case neither player wins. Notions of plays and strategies in the reachability game follow from the definitions for mean-payoff games, however we extend plays to include finite paths that end in $T_\exists \cup T_\forall$.

10.1 Undecidability of Mean-Payoff Games with Partial Observation

MPGs with partial observation were extensively studied in [DDG+10]. In that paper the authors show that, with the mean payoff condition defined using $\underline{\mathsf{MP}}$ and $>$, determining whether Eve has a winning strategy is undecidable and when defined using $\overline{\mathsf{MP}}$ and $\geq$, strategies with infinite memory may be necessary. The analogous questions using $\underline{\mathsf{MP}}$ and $\geq$ were left open. In this section we answer these questions, showing that both results still hold.

Let us start by formally stating the results already known from [DDG+10].

Proposition 10.1 (From [DDG+10])**.**

- *There are $\overline{\mathsf{MP}}$ games with partial observation for which Eve requires infinite memory observation-based strategies to ensure a non-negative mean-payoff value (i.e. ≥ 0).*
- *Determining if Eve has an observation-based strategy to ensure a positive mean-payoff value (i.e. > 0), given an $\overline{\mathsf{MP}}$ or $\underline{\mathsf{MP}}$ game with partial observation, is undecidable even for blind games.*
- *Determining if Eve has an observation-based strategy to ensure a non-negative mean-payoff value (i.e. ≥ 0), given an $\overline{\mathsf{MP}}$ game with partial observation, is undecidable.*
- *Determining if Eve has a finite-memory observation-based strategy to ensure a non-negative mean-payoff value (i.e. ≥ 0), given a $\overline{\mathsf{MP}}$ or $\underline{\mathsf{MP}}$ game with partial observation, is undecidable even for blind games.*

■

We will now argue that infinite memory is also necessary for Eve in some $\underline{\mathsf{MP}}$ games. Furthermore, as a corollary of a construction we present later in this chapter, we obtain undecidability for general $\underline{\mathsf{MP}}$ games and ≥ 0.

Proposition 10.2. *There are $\underline{\mathsf{MP}}$ games with partial observation for which Eve requires infinite memory observation-based strategies to ensure a non-negative mean-payoff value.* ■

Proof of Proposition 10.2. Consider the game $\mathcal{G}$ in Figure 10.1. We will show that Eve has an infinite memory observation-based strategy to win this game, but no finite memory observation-based strategy.

Consider the strategy that plays (regardless of location) $aba^2ba^3ba^4b\ldots$ As b is played infinitely often in this strategy, the only concrete paths consistent with this strategy are $\pi = q_0 q_1^\omega$ and $\pi = q_0 \cdot q_1^k \cdot q_2^l \cdot q_3^\omega$ for non-negative integers k, l. In the first case we see that $\frac{1}{n} w(\pi[..n]) \to 0$ as $n \to \infty$, and for all paths matching the second case we have $\frac{1}{n} w(\pi[..n]) \to 1$ as $n \to \infty$. Thus $\underline{\mathsf{MP}} \geq 0$ and so the strategy is winning.

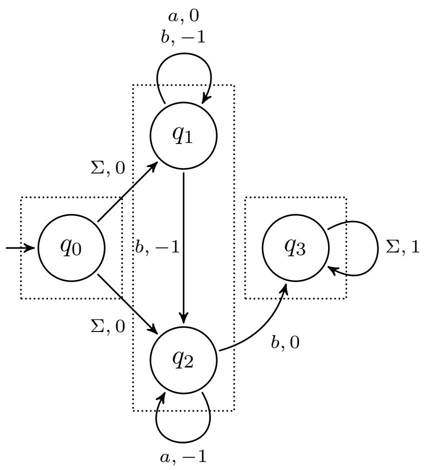

Figure 10.1: A limited-observation MPG in which Eve requires infinite memory to win.

Now suppose Eve has a finite memory observation-based winning strategy for $\mathcal{G}$. Consider the observation-based strategy of Adam that ensures the game remains in $\{q_1, q_2\}$. The resulting play can now be seen as choosing a word $w \in \{a, b\}^\omega$, but as Eve's strategy has finite memory, this word must be ultimately periodic, that is $w = w_0 \cdot v^\omega$ for words $w_0, v \in \{a, b\}^*$. But then Adam has a concrete winning strategy as follows. If w contains finitely many b's then Adam moves to q_2 on the final b and $\frac{1}{n}w(\pi[..n]) \to -1$ as $n \to \infty$. Otherwise Adam remains in q_1 and $\frac{1}{n}w(\pi[..n]) \to -\frac{m}{|v|}$ as $n \to \infty$ where m is the number of b's in v. □

Theorem 10.1. *Determining whether Eve has an observation-based strategy to ensure non-negative mean-payoff value, given a* $\underline{\mathsf{MP}}$ *game with partial observation, is undecidable.* ■

The proof of this result is based on a similar construction to the one used in the proof of Theorem 10.4, so we defer it to Section 10.3.1.

10.2 Strategy Transfer from an Extended Weighted Unfolding

In this section, we will construct a reachability game from an MPG with limited observation in which winning strategies for either player are sufficient (but not necessary) for observation-based finite-memory winning strategies in the original MPG. It is known that if a finite-memory winning strategy exists for a player in an MPG with partial observation, then it is winning for both versions of the mean-payoff function, *i.e.* $\underline{\mathsf{MP}}$ and $\overline{\mathsf{MP}}$. For consistency with the previous section, let us assume the $\liminf$ mean payoff henceforth.

Let us fix a mean-payoff game with limited observation $\mathcal{G} = (Q, q_I, \Sigma, \Delta, \mathsf{Obs}, w)$. We will define a reachability game on the weighted unfolding of $\mathcal{G}$.

Let $\mathcal{F}$ be the set of functions $f : Q \to \mathbb{Z} \cup \{+\infty, \bot\}$. Our intention is to use functions in $\mathcal{F}$ to keep track of the minimum weight of all concrete paths ending in the given state. A function value of $\bot$ indicates that the given state is not in the current observation, and a function value of $+\infty$ is used to indicate (to Eve) that the token is not located at such a state. Intuitively, $+\infty$ will allow our reachability winning condition to include games where Adam wins by ignoring paths going through the given state. The *support* of f is $\mathsf{supp}(f) = \{q \in Q \mid f(q) \neq \bot\}$. We say that $f' \in \mathcal{F}$ is a σ-successor of $f \in \mathcal{F}$ if:

- $\mathsf{supp}(f') \in \mathsf{Obs} \wedge \mathsf{supp}(f') \subseteq \mathsf{post}_\sigma(\mathsf{supp}(f))$; and
- for all $q \in \mathsf{supp}(f')$, $f'(q)$ is either $\min\{f(q') + w(q', \sigma, q) \mid q' \in \mathsf{supp}(f) \wedge (q', \sigma, q) \in \Delta\}$ or $+\infty$.

We define a family of partial orders, $\preceq_k$ ($k \in \mathbb{N}$), on $\mathcal{F}$ by setting $f \preceq_k f'$ if $\mathsf{supp}(f) = \mathsf{supp}(f')$ and $f(q) + k \leq f'(q)$ for all $q \in \mathsf{supp}(f)$ (where $+\infty + k = +\infty$).

Denote by $\mathfrak{F}_\mathcal{G}$ the set of all sequences $f_0\sigma_0 f_1 \ldots \sigma_{n-1} f_n \in (\mathcal{F} \cdot \Sigma)^* \cdot \mathcal{F}$ such that for all $0 \leq i < n$, f_{i+1} is a σ_i-successor of f_i. Observe that for each function-action sequence $\varrho = f_0\sigma_0 \ldots f_n \in \mathfrak{F}_\mathcal{G}$ there is a unique abstract path $\mathsf{supp}(\varrho) = o_0\sigma_0 \ldots o_n$ such that $o_i = \mathsf{supp}(f_i)$ for all i. Conversely, for each abstract path $\psi = o_0\sigma_0 \ldots o_n$ there may be many corresponding function-action sequences in $\mathsf{supp}^{-1}(\psi)$.

Reachability game. The *reachability game associated with* $\mathcal{G}$, *i.e.* $\Gamma_\mathcal{G} = (\Pi_\mathcal{G}, \Sigma, f_I, \delta, T_\exists, T_\forall)$, is formally defined as follows:

- The function $f_I \in \mathcal{F}$ is such that $f(q) = 0$ if $q = q_I$ and $f(q) = \bot$ otherwise.
- $\Pi_\mathcal{G}$ is the subset of $\mathfrak{F}_\mathcal{G}$ where for all $f_0\sigma_0 f_1 \ldots \sigma_{n-1} f_n \in \Pi_\mathcal{G}$ we have $f_0 = f_I$ and for all $0 \leq i < j < n$ we have $f_i \not\preceq_0 f_j$ and $f_j \not\preceq_1 f_i$;
- δ is the natural transition relation, that is, if x and $x \cdot \sigma \cdot f$ are elements of $\Pi_\mathcal{G}$ then $(x, \sigma, x \cdot \sigma \cdot f) \in \delta$;
- $T_\exists$ is the set of all $f_0\sigma_0 f_1 \ldots \sigma_{n-1} f_n \in \Pi_\mathcal{G}$ such that for some $0 \leq i < n$ we have $f_i \preceq_0 f_n$; and
- $T_\forall$ is the set of all $f_0\sigma_0 f_1 \ldots \sigma_{n-1} f_n \in \Pi_\mathcal{G}$ such that for some $0 \leq i < n$ we have $f_n \preceq_1 f_i$ and $f_i(q) \neq +\infty$ for some $q \in \mathsf{supp}(f_i)$.

Note that the directed graph defined by $\Pi_\mathcal{G}$ and δ is a tree, but not necessarily finite. To gain some intuition about $\Gamma_\mathcal{G}$, let us say an abstract cycle ϱ is *good* if there exists $f_0\sigma_0 \ldots f_n \in \mathsf{supp}^{-1}(\varrho)$ such that $f_i(q) \neq +\infty$ for all q and all i and $f_0 \preceq_0 f_n$. Let us say ϱ is *bad* if there exists $f_0\sigma_0 \ldots f_n \in \mathsf{supp}^{-1}(\varrho)$ such that $f_0(q) \neq +\infty$ for some $q \in \mathsf{supp}(f_0)$ and $f_n \preceq_1 f_0$. Then it is not difficult to see that $\Gamma_\mathcal{G}$ is essentially an abstract cycle-forming game played on $\mathcal{G}$ which is winning for Eve if a good abstract cycle is formed and winning for Adam if a bad abstract cycle is formed.

Our main result for this section is the following:

Theorem 10.2. *Let $\mathcal{G}$ be an MPG with limited observation and let $\Gamma_{\mathcal{G}}$ be the associated reachability game. If Adam (Eve) has a winning strategy in $\Gamma_{\mathcal{G}}$ then (s)he has a finite-memory observation-based winning strategy in $\mathcal{G}$.* ■

The idea behind the strategy for the mean-payoff game is straightforward. If Eve wins the reachability game then she can transform her strategy into one that plays indefinitely by returning, whenever the play reaches $T_\exists$, to the natural previous position—namely the position that witnesses the membership of $T_\exists$. By continually playing her winning strategy in this way Eve perpetually completes good abstract cycles and this ensures that all concrete paths consistent with the play have non-negative mean-payoff value. Likewise if Adam has a winning strategy in the reachability game, he can continually play his strategy by returning to the natural position whenever the play reaches $T_\forall$. By doing this he perpetually completes bad abstract cycles and this ensures that there is at least one concrete path consistent with the play that has strictly negative mean-payoff value.

The following definitions will be used throughout this section.

Definition (Proper σ-successor)**.** A *proper* σ-successor of a function f is a $\preceq_0$-minimal σ-successor of f. ■

Note that σ-successors that are not proper assign $+\infty$ to some state.

We observed earlier that for an abstract play $\psi = o_0\sigma_0 o_1 \ldots$ there may be many function-action sequences in $\mathsf{supp}^{-1}(\psi)$. However, for every f with $\mathsf{supp}(f) = o_0$ there is a unique $\preceq_0$-minimal sequence in $\mathsf{supp}^{-1}(\psi)$ starting at f obtained by taking proper successors with appropriate supports. We denote this sequence by $\xi(\psi, f)$.

For convenience given a finite function-action sequence $\varrho = f_0\sigma_0 \ldots f_n$, let f_ϱ denote f_n.

We will repeatedly use the next result which follows by induction immediately from the definition of a σ-successor.

Lemma 10.2. *Let $\varrho = f_0\sigma_0 \ldots f_n \in \mathfrak{F}_G$ be a sequence such that f_{i+1} is a proper σ_i-successor of f_i, for all i. Then for all $q \in \mathsf{supp}(f_n)$*

$$f_n(q) = \min\{f_0(\pi[0]) + w(\pi) \mid \pi \in \mathsf{obs}^{-1}(\mathsf{supp}(\varrho)) \text{ and } \pi[n] = q\}.$$

■

The following simple facts about $\preceq_n$ will also be useful:

Lemma 10.3. *For any $f_1, f_2 \in \mathcal{F}$ with $f_1 \preceq_k f_2$:*

(i) For all $k' \leq k$, $f_1 \preceq_{k'} f_2$,

(ii) For all $k' \geq 0$, if $f_2 \preceq_{k'} f_3$ for some $f_3 \in \mathcal{F}$ then $f_1 \preceq_{k+k'} f_3$, and

(iii) If f_1' is a proper σ-successor of f_1 and f_2' is a σ-successor of f_2 with $\mathsf{supp}(f_2') = \mathsf{supp}(f_1')$, then $f_1' \preceq_k f_2'$.

■

Proof. (i) and (ii) are trivial. For (iii), let $d_{i,j} = w(q_i, \sigma, q_j)$ for $q_i \in \mathsf{supp}(f_1)$ and $q_j \in \mathsf{supp}(f_1')$ where such a transition exists and $+\infty$ otherwise. We now observe that as f_1' is $\preceq_0$-minimal, $f_1'(q_j)$ can be defined as $\min\{f_1(q_i)+d_{i,j}|q_i \in \mathsf{supp}(f_1)\}$ for all $q_j \in \mathsf{supp}(f_1')$. As $f_1(q_i) \le f_2(q_i) - k$ for any $q_i \in \mathsf{supp}(f_1)$, it follows that

$$f_1'(q_j) \le \min\{f_2(q_i) + d_{i,j} \mid q_i \in \mathsf{supp}(f_1)\} - k \le f_2'(q_j) - k,$$

where the second inequality follows from the definition of a σ-successor. Thus $f_1' \preceq_k f_2'$. □

Although the following results are not used until Section 10.3.2, they already give an intuition toward the correctness of the strategies described above. In words, we will show that repeating good cycles is itself, in some sense, good, while repeating bad ones is bad.

Lemma 10.4. *Let ϱ be an abstract cycle.*

(i) If ϱ is good (bad) then an interleaving of ϱ with another good (bad) cycle is also good (bad).

(ii) If ϱ is good then for all k and all concrete cycles $\pi \in \mathsf{obs}^{-1}(\varrho^k)$, $w(\pi) \ge 0$.

(iii) If ϱ is bad then $\exists k \ge 0, \pi \in \mathsf{obs}^{-1}(\varrho^k)$ such that $w(\pi) < 0$.

■

Proof. (i) follows from Lemma 10.3. For (ii), let $f_0\sigma_0 \dots f_n \in \mathsf{supp}^{-1}(\varrho)$ be such that $f_i(q) \ne +\infty$ for all i and q and $f_0 \preceq_0 f_n$. In particular this means that f_{i+1} is a proper σ_i-successor of f_i. Now fix k and let $\chi \in \mathsf{obs}^{-1}(\varrho^k)$ be a concrete cycle. From Lemma 10.2 we have, for all $0 \le i < k$,

$$w(\chi[ni..n(i+1)]) \ge f_n(\chi[n(i+1)]) - f_0(\chi[ni])$$

and

$$f_n(\chi[n(i+1)]) - f_0(\chi[ni]) \ge f_0(\chi[n(i+1)]) - f_0(\chi[ni]).$$

Hence

$$w(\chi) = \sum_{i=1}^{k} w(\chi[ni..n(i+1)]) \ge f_0(\chi[nk]) - f_0(\chi[0]) = 0.$$

(iii) Let $f_0\sigma_0 \dots f_n \in \mathsf{supp}^{-1}(\varrho)$ and $q_0 \in \mathsf{supp}(f_0)$ be such that $f_0(q_0) \ne +\infty$ and $f_n \preceq_1 f_0$. It follows that $f_n(q_0) < +\infty$. From the definition of a σ-successor, it follows that there exists $r \in \mathsf{supp}(f_{n-1})$ such that $f_{n-1}(r) < +\infty$, and there is an edge from r to q_0 with weight $f_n(q_0) - f_{n-1}(r)$. Proceeding this way inductively we find there is a $q_1 \in \mathsf{supp}(f_0)$ with $f_0(q_1) < +\infty$ and a concrete path $\pi_0 \in \mathsf{obs}^{-1}(\varrho)$ from q_1 to q_0 with $w(\pi_0) = f_n(q_0) - f_0(q_1)$. As $f_0(q_1) < +\infty$ and $f_n \preceq_1 f_0$ we have $f_n(q_1) \le f_0(q_1) - 1 < +\infty$. Repeating the argument yields a sequence of states $q_0, q_1, \dots$ such that there is a concrete path $\pi_i \in \mathsf{obs}^{-1}(\varrho)$ from q_{i+1} to q_i with

$$w(\pi_i) = f_n(q_i) - f_0(q_{i+1}) \le f_0(q_i) - f_0(q_{i+1}) - 1.$$

As Q is finite it follows that there exists $i < j$ such that $q_i = q_j$. Then the concrete path $\pi = \pi_j \cdot \pi_{j-1} \cdots \pi_{i+1} \in \mathsf{obs}^{-1}(\varrho^{j-i})$ is a concrete cycle with

$$w(\pi) = \sum_{k=i+1}^{j} w(\pi_k) \le f_0(q_i) - f_0(q_j) - (j - i) < 0.$$

□

Corollary 9. *No cyclic permutation of a good abstract cycle is bad.* ■

10.2.1 Strategy transfer for Eve

We note that, as a play prefix in $\Gamma_{\mathcal{G}}$ is completely described by the last state in the sequence, it suffices to consider positional strategies for both players.

Let us first assume that Eve has a (positional) winning strategy in $\Gamma_{\mathcal{G}}$, $\lambda : \Pi_{\mathcal{G}} \to \Sigma$. Let Γ_λ denote the restriction of $\Gamma_{\mathcal{G}}$ to plays consistent with λ, and let $M = \Pi'_\lambda$, the corresponding restriction of $\Pi_{\mathcal{G}}$. From Lemma 10.9, M is finite. We will define a strategy with memory $|M|$, λ^*, for Eve in $\mathcal{G}$. Given a memory state $\mu = f_0\sigma_0 f_1 \cdots \sigma_{n-1} f_n \in M$ let

$$\mu' = \begin{cases} \text{the proper prefix of } \mu \text{ such that } f_{\mu'} \preceq_0 f_\mu & \text{if } \mu \in T_\exists \\ \mu & \text{otherwise.} \end{cases}$$

The initial memory state is $\mu_0 := f_I$. We define the output function $\alpha_o : M \times \mathsf{Obs} \to \Sigma$ as $\alpha_o(\mu, o) = \lambda(\mu')$. Finally we define the update function $\alpha_u : M \times \mathsf{Obs} \to M$ as $\alpha_u(\mu, o) = \mu' \cdot \lambda(\mu') \cdot f$ where f is the proper $\lambda(\mu')$-successor of $f_{\mu'}$ with $\mathsf{supp}(f) = o$. Observe that we maintain the invariant that the current observation is $\mathsf{supp}(f_\mu)$, consequently the Obs input to α_o is not used.

We will show shortly that λ^* is a winning strategy for Eve in $\mathcal{G}$. First we require some definitions and a result about finite prefixes of plays consistent with λ^*. Given a play $\varrho = o_0\sigma_0 o_1 \ldots$ consistent with λ^*, let μ_i^ϱ denote the i-th memory state reached in the generation of ϱ. That is, $\mu_0^\varrho = \mu_0$ and $\mu_{i+1}^\varrho = \alpha_u(\mu_i^\varrho, \sigma_i)$; so $\alpha_o(\mu_i^\varrho, o_i) = \sigma_i$. Recall the definition of $\xi(\cdot,\cdot)$ from the start of the section. For convenience, let $\xi_i^\varrho = \xi(\varrho[..i], f_I)$.

Lemma 10.5. *Let $\varrho \in \mathsf{obs}(\mathsf{Plays}(\mathcal{G}))$ be a play consistent with λ^*. Then for all i, $f_{\mu_i^\varrho} \preceq_0 f_{\xi_i^\varrho}$.* ■

Proof. We prove this by induction. Let $\varrho = o_0\sigma_0 o_1 \ldots$ For $i = 0$ we have

$$f_{\mu_0^\varrho} = f_{\mu_0} = f_I = f_{\xi(o_0, f_I)} = f_{\xi_i^\varrho}.$$

Now suppose $f_{\mu_i^\varrho} \preceq_0 f_{\xi_i^\varrho}$. Let $f' = f_{\xi_{i+1}^\varrho}$, so f' is the proper σ_i-successor of $f_{\xi_i^\varrho}$ with $\mathsf{supp}(f') = o_{i+1}$. Assume first that $\mu_i^\varrho \notin T_\exists$. Then $\mu_{i+1}^\varrho = \alpha_u(\mu_i^\varrho, o_i) = \mu_i^\varrho \cdot \sigma_i \cdot f$, where f is the proper σ_i-successor of $f_{\mu_i^\varrho}$ with $\mathsf{supp}(f) = o_{i+1}$. Then, by Lemma 10.3 (iii) we have $f_{\mu_{i+1}^\varrho} = f \preceq_0 f'$.

Now assume $\mu_i^\varrho \in T_\exists$, and let μ' denote the proper prefix of μ_i^ϱ such that $f_{\mu'} \preceq_0 f_{\mu_i^\varrho}$. Then $\mu_{i+1}^\varrho = \alpha_u(\mu_i^\varrho, o_i) = \mu' \cdot \sigma_i \cdot f$ where f is the proper σ_i-successor of $f_{\mu'}$ with $\mathsf{supp}(f) = o_{i+1}$. From Lemma 10.3 (ii) we have $f_{\mu'} \preceq_0 f_{\xi_i^\varrho}$, so by Lemma 10.3 (iii) we have $f_{\mu_{i+1}^\varrho} = f \preceq_0 f'$ as required. □

We now proceed with the proof of strategy transfer for Eve.

Lemma 10.6. *Let $\mathcal{G}$ be a mean-payoff game with limited observation and let $\Gamma_{\mathcal{G}}$ be the associated reachability game. If Eve has a winning strategy in $\Gamma_{\mathcal{G}}$ then she has a finite memory winning strategy in $\mathcal{G}$.* ■

Proof. We will show that λ^* described above is a winning strategy for Eve. Let $\varrho = o_0\sigma_0 \cdots \in \mathsf{obs}(\mathsf{Plays}(\mathcal{G}))$ be any play consistent with λ^*. We will show that there exists a constant $\beta \in \mathbb{R}$ such that for all concrete paths $\pi \in \mathsf{obs}^{-1}(\varrho)$ and all $n \geq 0$, $w(\pi[..n]) \geq \beta$. It follows that $\underline{\mathsf{MP}}(\pi) \geq 0$, and so ϱ is winning for Eve.

Let $W = \{f_\mu(q) \mid \mu \in M, q \in Q, \text{ and } f_\mu(q) \neq \bot\}$. Note that W is finite because M and Q are finite, and non-empty because $f_{\mu_0}(q_I) = 0 \in W$. Let $\beta = \min W$. As $0 \in W$, $\beta < +\infty$.

As with Lemma 10.5, let $\xi_n^\varrho = \xi(\varrho[..n], f_I)$. As $\mathsf{supp}(f_I) = \{q_I\}$ and $f_I(q_I) = 0$, Lemma 10.2 implies for all $q \in \mathsf{supp}(f_{\xi_n^\varrho})$, $f_{\xi_n^\varrho}(q) \neq +\infty$. Hence, for all concrete paths $\pi \in \mathsf{obs}^{-1}(\varrho)$ we have:

$$\begin{array}{rcll} w(\pi[..n]) & \geq & f_{\xi_n^\varrho}(\pi[n]) - f_I(\pi[0]) & \text{from Lemma 10.2} \\ & = & f_{\xi_n^\varrho}(\pi[n]) & \\ & \geq & f_{\mu_n^\varrho}(\pi[n]) & \text{from Lemma 10.5} \\ & \geq & \beta & \text{as required.} \end{array}$$

□

10.2.2 Strategy transfer for Adam

To complete the proof of Theorem 10.2, we now show how to transfer a winning strategy for Adam in $\Gamma_\mathcal{G}$ to a winning strategy in $\mathcal{G}$. So let us assume $\lambda : \Pi_\mathcal{G} \times \Sigma \to \Pi_\mathcal{G}$ is a (positional) winning strategy for Adam in $\Gamma_\mathcal{G}$. The finite-memory observation-based strategy for Adam is similar to that for Eve in that it perpetually plays λ, returning to a previous position whenever the play reaches $T_\forall$. However, the proof of correctness is more intricate because we need to handle the $+\infty$ function values.

Formally, the finite-memory strategy λ^* is given as follows. As before, let $M = \Pi'_\lambda$ and $\mu_0 = f_I$. Given $\mu \in M$, let

$$\mu' = \begin{cases} \text{the proper prefix of } \mu \text{ such that } f_\mu \preceq_1 f_{\mu'} & \text{if } \mu \in T_\forall \\ \mu & \text{otherwise.} \end{cases}$$

The output function $\alpha_o : M \times \mathsf{Obs} \times \Sigma \to \mathsf{Obs}$ is defined as: $\alpha_o(\mu, o, \sigma) = \mathsf{supp}(\lambda(\mu', \sigma))$. The update function $\alpha_u : M \times \mathsf{Obs} \times \Sigma \to M$ is defined as: $\alpha_u(\mu, o, \sigma) = \mu' \cdot \sigma \cdot \lambda(\mu', \sigma)$. Note that as the current observation is stored in the memory state, the Obs input to α_o and α_u is obsolete.

To show that λ^* is winning for Adam in $\mathcal{G}$ we require an analogue to Lemma 10.5. Given a play $\varrho = o_0\sigma_0 \ldots$ in $\mathcal{G}$ consistent with λ^*, let μ_i^ϱ be the i-th memory state reached in the generation of ϱ. That is, $\mu_0^\varrho = \mu_0$ and $\mu_{i+1}^\varrho = \alpha_u(\mu_i^\varrho, o_i, \sigma_i)$, so $\alpha_o(\mu_i^\varrho, o_i, \sigma_i) = o_{i+1}$. Let r_i^ϱ denote the number of times the memory is "reset" in the first i steps. That is, $r_0^\varrho = 0$, and if $\mu_i^\varrho \in T_\forall$ then $r_{i+1}^\varrho = 1 + r_i^\varrho$, otherwise $r_{i+1}^\varrho = r_i^\varrho$. Rather than relate $f_{\mu_i^\varrho}$ with functions in $\xi(\varrho, f_I)$, we need to consider more general sequences which are $\preceq_0$-minimal after some point. Let us say a function-action sequence $f_0\sigma_0 f_1 \cdots \in \mathfrak{F}_\mathcal{G}$ is *ultimately proper from* k if for all $i \geq k$, f_{i+1} is a proper σ_i-successor of f_i.

Lemma 10.7. *Let $\varrho \in \mathsf{obs}(\mathsf{Plays}(\mathcal{G}))$ be a play consistent with λ^* and $\zeta = z_0\sigma_0 z_1 \cdots \in \mathsf{supp}^{-1}(\varrho)$ be ultimately proper from k. If $z_k \preceq_r f_{\mu_k^\varrho}$ for some r, then for all $i \geq k$, $z_i \preceq_{r'_i} f_{\mu_i^\varrho}$ where $r'_i = r + r_i^\varrho - r_k^\varrho$.* ■

Proof. We prove this by induction on i. For $i = k$ the result clearly holds. Now suppose $i \geq k$ and $z_i \preceq_{r'_i} f_{\mu_i^\varrho}$ where $r'_i = r + r_i^\varrho - r_k^\varrho$. We consider two cases depending on whether $\mu_i^\varrho \in T_\forall$. If $\mu_i^\varrho \notin T_\forall$ then $\mu_{i+1}^\varrho = \mu_i^\varrho \cdot \sigma_i \cdot f$ where f is a σ_i-successor of $f_{\mu_i^\varrho}$ with $\mathsf{supp}(f) = o_{i+1}$, and $r_{i+1}^\varrho = r_i^\varrho$. Then, by Lemma 10.3 (iii) we have $z_{i+1} \preceq_{r'_i} f = f_{\mu^\varrho_{i+1}}$ and $r'_i = r + r^\varrho_i - r^\varrho_k = r + r^\varrho_{i+1} - r^\varrho_k = r'_{i+1}$ as required.

Otherwise if $\mu^\varrho_i \in T_\forall$, let μ' be the proper prefix of μ^ϱ_i such that $f_{\mu^\varrho_i} \preceq_1 f_{\mu'}$. From Lemma 10.3 (ii) we have $z_i \preceq_{r'_i+1} f_{\mu^\varrho_{i+1}}$. We also have $\mu_{i+1} = \mu' \cdot \sigma_{i+1} \cdot f$ where f is a σ_{i+1}-successor of $f_{\mu'}$ with $\mathsf{supp}(f) = o_{i+1}$. So by Lemma 10.3 (iii) we have $z_{i+1} \preceq_{1+r'_i} f = f_{\mu^\varrho_{i+1}}$, and as $r'_i+1 = r+r^\varrho_i - r^\varrho_k + 1 = r + r^\varrho_{i+1} - r^\varrho_k = r'_{i+1}$ the result holds for $i+1$. □

We now show how to transfer strategies of Adam.

Lemma 10.8. *Let $\mathcal{G}$ be a mean-payoff game with limited observation and let $\Gamma_\mathcal{G}$ be the associated reachability game. If Adam has a winning strategy in $\Gamma_\mathcal{G}$ then he has a finite memory winning strategy in $\mathcal{G}$.* ■

Proof. We will show that the strategy λ^* constructed above is winning for Adam. Let $\varrho = o_0\sigma_0\ldots$ be any play consistent with λ^*. As M is finite, there exists $\mu \in M$ and an infinite set $\mathcal{I} \subseteq \mathbb{N}$ of indices such that for all $i \in \mathcal{I}$, $\mu^\varrho_i = \mu$. We will show that this implies there exists $\pi \in \mathsf{obs}^{-1}(\varrho)$ such that $\overline{\mathsf{MP}}(\pi) < 0$. As $\overline{\mathsf{MP}}(\pi) \geq \underline{\mathsf{MP}}(\pi)$ the result follows. For convenience, given $n \in \mathbb{N}$, let $\mathsf{s}_\mathcal{I}(n) = \min\{i \in \mathcal{I} \mid i > n\}$. Also, let $o = \{q \in \mathsf{supp}(f_\mu) \mid f_\mu(q) \neq +\infty\}$. Note that from the definition of $T_\forall$ it follows that o is non-empty.

We will use a function-action sequence to find a concrete path where the weights of the prefixes can be identified and seen to be strictly decreasing. In Lemma 10.6 the sequence $\xi(\varrho, f_I)$ fulfilled this role. However, to handle $+\infty$ values, which correspond to irrelevant paths, here we require a sequence more complex than $\xi(\varrho, f_I)$. The sequence we construct will be *piecewise proper* in the sense that for all $i \in \mathcal{I}$ the sequence will consist of proper successors in the interval $[i, \mathsf{s}_\mathcal{I}(i))$. When the sequence reaches an element of $\mathcal{I}$ we "reset" the values of the vertices not in o to $+\infty$. More formally, the required sequence, $\zeta = z_0\sigma_0 \cdots \in \mathsf{supp}^{-1}(\varrho)$, is constructed inductively as follows. Initially, let $z_0 = z'_0 = f_I$. For $i \geq 0$, let z'_{i+1} be the proper σ_i-successor of z_i with $\mathsf{supp}(z_{i+1}) = o_{i+1}$. If $i \notin \mathcal{I}$ then $z_i = z'_i$. Otherwise,

$$z_i(q) = \begin{cases} +\infty & \text{if } q \notin o \\ z'_i(q) & \text{otherwise.} \end{cases}$$

We claim that for all $i \in \mathbb{N}$: $z_i \preceq_{r^\varrho_i} f_{\mu^\varrho_i}$. From Lemma 10.7 it follows that we only need to show that for all $i \in \mathcal{I}$: $z_i \preceq_{r^\varrho_i} f_\mu$. Induction and Lemma 10.7 imply that for all $i \in \mathcal{I}$ we have $z'_i \preceq_{r^\varrho_i} f_\mu$. As z_i differs from z'_i only on states where f_μ is equal to $+\infty$, we therefore have $z_i \preceq_{r^\varrho_i} f_\mu$ as required.

We will now show that there is an infinite concrete path $q_0\sigma_0\ldots$ consistent with ϱ such that $q_i \in o$ for all $i \in \mathcal{I}$. To do this we will show for any $i \in \mathcal{I}$ and any $q \in o$ there is a concrete path, consistent with $\varrho[i..\mathsf{s}_\mathcal{I}(i)]$, that ends in q and starts at some state in o. The result then follows by induction. Let us fix $i \in \mathcal{I}$, $q \in o$, and let $j = \mathsf{s}_\mathcal{I}(i)$. As $z'_j \preceq_{r^\varrho_j} f_\mu$, we have that $z'_j(q) \neq +\infty$. From Lemma 10.2, there is a concrete path $\pi = q_0\sigma_0\ldots q_n$ from $q_0 \in o_i$ ending at $q_n = q$ such that $z'_j(q) = z_i(q_0) + w(\pi)$. As $z'_j(q) \neq +\infty$ it follows that $z_i(q_0) \neq +\infty$, and as $z_i(q_0) = +\infty$ if and only if $f_\mu(q_0) = +\infty$, it follows that $q_0 \in o$. Note that Lemma 10.2 implies for all $k \leq |\pi|$: $w(\pi[..k]) = z'_{i+k}(q_k) - z_i(q_0) = z_{i+k}(q_k) - z_i(q_0)$. In particular $w(\pi) = z_j(q) - z_i(q_0)$.

Now let $\pi = q_0\sigma_0\ldots$ be the infinite path implied by the above construction and for convenience for $i \in \mathcal{I}$ let $\pi_i = q_i\sigma_i\ldots q_j$ where $j = \mathsf{s}_{\mathcal{I}}(i)$. To show $\overline{\mathsf{MP}}(\pi) < 0$ we need to show $\limsup_{n\to\infty} \frac{1}{n} w(\pi[..n]) < 0$. To prove this, we will show there exists a constant $\beta < 0$ such that for all sufficiently large n: $w(\pi[..n]) \leq \beta \cdot n$.

For convenience, let $i_0 = \min \mathcal{I}$ and let $i_n = \max\{i \in \mathcal{I} \mid i \leq n\}$. From Lemma 10.2 and the construction of π_i we have for all n:

$$\begin{aligned}
w(\pi[..n]) &= w(\pi[..i_0]) + w(\pi[i_n..n]) + \sum_{\substack{i\in\mathcal{I}\\ i<n}} w(\pi_i)\\
&= w(\pi[..i_0]) + (z_n(q_n) - z_{i_n}(q_{i_n}))\\
&\quad + \sum_{\substack{i\in\mathcal{I}\\ i<n}} z_{\mathsf{s}_{\mathcal{I}}(i)}(q_{\mathsf{s}_{\mathcal{I}}(i)}) - z_i(q_i)\\
&= w(\pi[..i_0]) + z_n(q_n) - z_{i_0}(q_{i_0})\\
&\leq w(\pi[..i_0]) + f_{\mu_n^\varrho}(q_n) - r_n^\varrho - z_{i_0}(q_{i_0})
\end{aligned}$$

There are only finitely many values for $f_{\mu_n^\varrho}(q_n)$ and from Lemma 10.9, $r_n^\varrho \geq \lfloor \frac{n}{N} \rfloor$. Hence

$$w(\pi[..n]) \leq \alpha - \beta' \cdot n$$

for constants α and $\beta' > 0$. Thus there exists $\beta < 0$ such that for sufficiently large n we have $w(\pi[..n]) \leq \beta \cdot n$. Hence $\underline{\mathsf{MP}}(\pi) \leq \overline{\mathsf{MP}}(\pi) < 0$. ☐

The finiteness of the size of the memory required for this strategy follows from the following result.

Lemma 10.9. *If λ is a winning strategy for Adam or Eve in $\Gamma_{\mathcal{G}}$, then there exists $N \in \mathbb{N}$ such that for all plays π consistent with λ, $|\pi| \leq N$.* ■

Proof. Let Γ_λ be the restriction of $\Gamma_{\mathcal{G}}$ to plays that are consistent with λ. Suppose there is no bound on the length of paths in Γ_λ. As $\Gamma_{\mathcal{G}}$, and hence Γ_λ, is acyclic, it follows that Γ_λ contains infinitely many states. However, as $\Gamma_{\mathcal{G}}$ is finitely-branching, it follows from König's lemma that there exists an infinite path in Γ_λ. As this path is not winning for either player and it is consistent with λ, this contradicts the fact that λ is a winning strategy. ☐

10.3 Decidable Classes of MPGs with Limited Observation

10.3.1 Forcibly terminating games

The reachability game defined in the previous section gives a sufficient condition for determining the winner in an MPG with limited observation. However, as there may be plays where no player wins, such games are not necessarily determined. The first subclass of MPGs with limited observation we investigate is the class of games where the associated reachability game is determined.

Definition (Forcibly terminating games)**.** An MPG with limited observation is *forcibly terminating* if in the corresponding reachability game $\Gamma_{\mathcal{G}}$ either Adam has a winning strategy to reach locations in $T_{\forall}$ or Eve has a winning strategy to reach locations in $T_{\exists}$. ■

It follows immediately from Theorem 10.2 that finite memory strategies suffice for both players in forcibly terminating games. Note that an upper bound on the memory required is the number of vertices in the reachability game restricted to a winning strategy, and this is exponential in N, the bound obtained in Lemma 10.9.

Theorem 10.3 (Finite-memory determinacy)**.** *One player always has a winning observation-based strategy with finite memory in a forcibly terminating MPG.* ■

We now consider the complexity of two natural decision problems associated with forcibly terminating games: the problem of recognizing if an MPG is forcibly terminating and the problem of determining the winner of a forcibly terminating game. Both results follow directly from the fact that we can accurately simulate a Turing machine with an MPG with limited observation.

Theorem 10.4. *Let $\mathcal{M}$ be a 4CM guaranteed to halt. Then there exists an MPG with limited observation $\mathcal{G}$, constructible in polynomial time, such that Eve wins $\Gamma_{\mathcal{G}}$ if and only if $\mathcal{M}$ halts in the accept state and Adam wins $\Gamma_{\mathcal{G}}$ if and only if $\mathcal{M}$ halts in the reject state.* ■

Proof. Recall that a Minsky machine $\mathcal{M}$ consists of a finite set of control states S, an initial state $s_I \in S$, a final accepting state $s_A \in S$, a final rejecting state s_R, a set C of integer-valued counters and a finite set $\delta_{\mathcal{M}}$ of instructions manipulating the counters. The transitions relation $\delta_{\mathcal{M}}$ contains tuples $(s, instr, c, s')$ where $s, s' \in S$ are source and target states, respectively, and $instr \in \{inc, dec, 0?\}$ applies to counter $c \in C$.

Given a 4CM $\mathcal{M}$, we now show how to construct an MPG with limited observation $\mathcal{G}_{\mathcal{M}}$ in which Eve wins $\Gamma_{\mathcal{G}_{\mathcal{M}}}$ if and only if $\mathcal{M}$ has an accepting run, and Adam wins $\Gamma_{\mathcal{G}_{\mathcal{M}}}$ if and only if $\mathcal{M}$ has a rejecting run. Plays in $\mathcal{G}_{\mathcal{M}}$ correspond to executions of $\mathcal{M}$. As we will see, the tricky part is to make sure that zero-check instructions are faithfully simulated by one of the players. Initially, both players will be allowed to declare how many instructions the machine needs to execute in order to reach an accepting or rejecting state. Either player can bail out of this initial "pumping phase" and become the *Simulator*. The Simulator is then responsible for the faithful simulation of $\mathcal{M}$ and the opponent will be monitoring the simulation and punish him if the simulation is not executed correctly. Let us now go into the details.

Control structure. First, the control structure of the machine $\mathcal{M}$ is encoded in the observations of our game, *i.e.* to each location of the machine, there will correspond at most three observations in the game. We require two copies of each such observation since, in order to punish Adam or Eve (whoever plays the role of Simulator), existential and universal gadgets have to be set up in a different manner. For technical reasons that will be made clear below, we also need two additional observations. Formally, the observation set in our game contain observations $\{b^+, b^0, b^-\}$, $\{a^+, a^-\}$ and $\{q_I\}$, which do not correspond

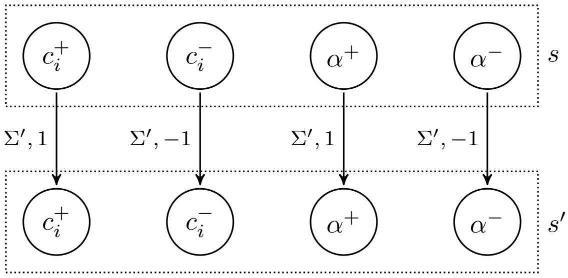

Figure 10.2: Observation gadget for (s, inc, c_i, s') instruction. For $(s, q) \in Q$ only the q component is shown.

to instructions from the 4CM but they are used in gadgets that will make sure that zero tests are faithfully executed.

Counter values. Second, the values of counters will be encoded using the weights of traversed edges that reach designated states. We will associate to each observation (so to each location in the 4CM) two states for each counter: c_i^+ and c_i^-, $i \in \{1, 2, 3, 4\}$. Intuitively, an abstract path, corresponding to the simulation of a run of the machine, will encode the value of counter i, at each step, as the weight of the shortest suffix from the initial pumping gadget to c_i^+.

$\mathcal{G}_{\mathcal{M}}$ starts in $\{\{q_I\}\}$ and Δ contains σ-transitions (for all $\sigma \in \Sigma'$) from q_I to b^+, b^0, b^-. This observation represents the pumping phase of the simulation. From here each player will be allowed to declare how many steps they require to reach a halting state that will accept or reject. If Adam bails, we go to the initial instruction of $\mathcal{M}$ on the universal side of the construction ($s_I^\forall$), if Eve does so then we go to the analogue in the existential side ($s_I^\exists$). Σ' contains a symbol `bail` which represents Eve choosing to leave the gadget and try simulating an accepting run of $\mathcal{M}$, that is $\Delta \ni (b^+, \texttt{bail}, (s_I^\exists, \alpha^-))$, $(b^-, \texttt{bail}, (s_I^\exists, \alpha^+))$, $(b^0, \texttt{bail}, (s_I^\exists, c))$ where $c \in \{c_i^+, c_i^-\}$ for all i. For all other actions in Σ', self-loops are added on the states b^+, b^0, b^- with weights $+1, 0, -1$ respectively. Meanwhile, Adam is able to exit the gadget at any moment—via non-deterministic transitions $(b^+, \sigma, (s_I^\forall, \alpha^-))$, $(b^-, \sigma, (s_I^\forall, \alpha^+))$, $(b^0, \sigma, (s_I^\forall, c))$ where $c \in \{c_i^+, c_i^-\}$ for all i and $\sigma \in \Sigma' \setminus \{\texttt{bail}\}$—to the universal side of the construction, *i.e.* he will try to simulate a rejecting run of the machine. Bailing transitions (transitions going to states $(s_I^\exists, \cdot)$ or $(s_I^\forall, \cdot)$) have weight 0.

Note that after these initial transitions the simulated value of all the counters is 0. Indeed, this corresponds to the beginning of a simulation of $\mathcal{M}$ starting from configuration (s_I, v) where $v(c) = 0$ for all $c \in C$.

Counter increments & decrements. Let us now explain how Eve simulates increments of counter values using this encoding (decrements are treated similarly). The gadget we explain below actually works the same in both sides of the construction, *i.e.* the universal and existential gadgets for increments and decrements are identical. For that, consider Figure 10.2, the upper part of the figure is related to the location (instruction) s of $\mathcal{M}$, while the bottom part is related to the location s' of $\mathcal{M}$, and assume that $(s, inc, c_i, s') \in \delta_{\mathcal{M}}$.

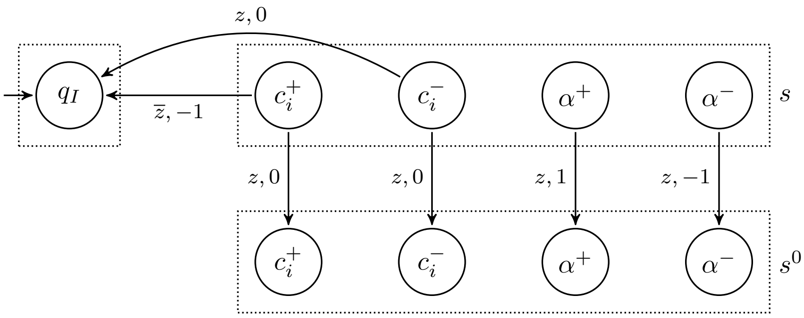

Figure 10.3: Existential observation gadget for $(s, 0?, c_i, s^0)$ and (s, dec, c_i, s') instructions. Transitions to s'-observation not shown.

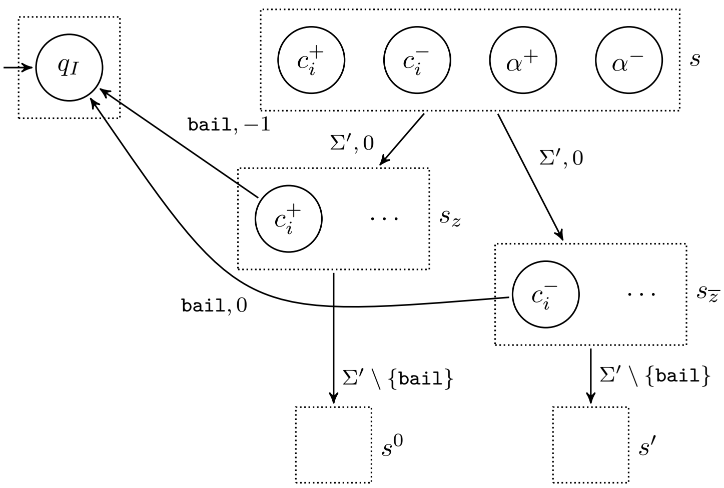

Figure 10.4: Universal observation gadget for $(s, 0?, c_i, s^0)$ and (s, dec, c_i, s') instructions. Transitions to s', s^0 observations are weighted as with the existential observation gadget.

As can be seen in the figure, the observation related to the instruction s contains: the states c_i^+, c_i^- are used to encode of the value of counter c_i. The additional states α^+, α^- are used to encode the number of steps in the simulation (again one positive ending in α^+ and one negative encoding in α^-). Now, let us consider the transitions of the gadget. The increment of the counter c_i from location s to location s' is encoded using the weights on the transitions that go from the observation s to the observation s'. As you can see, the weight on the edge between the copy of state c_i^+ of observation s to the copy of this state in observation s' is equal to $+1$, while the weight on the edge between the copy of state c_i^- of observation s to the copy of this state in observation s' is equal to -1. As you can see from the figure, when going from location s to location s', we also increment the additional counter that keeps track of the number of steps in the simulation of $\mathcal{M}$. As the machine is deterministic there is no choice for Eve in observation s, since only an increment can be executed, this is why, regardless of the action chosen from Σ', the same transition is taken.

Existential zero checks. Now, let us turn to the gadget of Figure 10.3, that is used to simulate zero-check instructions. We first focus on the case in which it is the duty of Eve to reveal if the counter has value zero or not, by forcibly choosing the next letter to play in $\{z, \overline{z}\} \subset \Sigma'$. In the observation that corresponds to the location s of $\mathcal{M}$, Eve decides to declare that the counter c_i is equal to zero (by issuing z) or not (by issuing $\overline{z}$), then Adam resolves non-determinism as follows. If Eve does not cheat then Adam should let the simulation to continue to either s^0 or s' depending on Eve's choice (the figure only depicts the branching to s^0, the branching to s' is similar). Now if Eve has cheated, then Adam should have a way to retaliate: we allow him to do so by branching to observation $\{q_I\}$ from state (s, c_i^-) with weight 0 in case z has been issued and the counter c_i is not equal to zero and with weight -1 in case $\overline{z}$ has been issued and the counter c_i is equal to zero. It should be clear that in both cases Adam closes a bad abstract cycle.

Universal zero checks. A similar trick is used for the gadget from Figure 10.4, where Adam is forced to simulate a truthful zero check or lose $\Gamma_{\mathcal{G}_\mathcal{M}}$. Since Adam can control non-determinism and not the action chosen, we have transitions going from $(s, \cdot)$ to states in both $(s_z, \cdot)$ and $(s_{\overline{z}})$ with weight 0 and all actions in Σ'. Eve is then allowed to branch back to $\{q_I\}$ as follows. If Adam does not cheat, then Eve will play any action in $\Sigma' \setminus \{\texttt{bail}\}$ and transitions, with weights similar to those used in the the existential check gadget, will take the play from $(s_z, \cdot)$ to $(s^0, \cdot)$ and from $(s_{\overline{z}}, \cdot)$ to $(s', \cdot)$. Now if Adam has cheated by taking the play to $(s_z, \cdot)$ when c_i was not zero, then Eve—by playing `bail`—can go from (s_z, c_i^+) to the initial observation with weight -1 and close a good abstract cycle. If Adam cheated by taking the play to $(s_{\overline{z}}, \cdot)$ when c_i was indeed zero, Eve can go (with the same action) from $(s_{\overline{z}}, c_i^-)$ to the initial observation with weight 0 again and close a good abstract cycle. Indeed, Adam can escape the zero check gadget by choosing a non-proper successor. We will shortly explain why this is not a viable option for him.

Stopping Simulator. It should be clear also from the gadgets, that the opponent of Simulator has no incentive to interrupt the simulation if there is no cheat. Doing so is actually beneficial to Simulator, who can get a function-action sequence which makes him win $\Gamma_{\mathcal{G}_\mathcal{M}}$.

Finally, Δ also contains self-loops at all $(s_A, \cdot)$ with all Σ' and with 0 weights and at all $(s_R, \cdot)$ with all Σ' and with -1 weights. Thus, if the play reaches the observation representing state s_F or s_R from $\mathcal{M}$ then Simulator will be able to force function-action sequences which allow him to win $\Gamma_{\mathcal{G}_\mathcal{M}}$.

Bound on the length of the simulation. All that is left is to explain the idea behind observation gadget $\{a^+, a^-\}$ and to show how we allow the opponent of Simulator to stop the simulated run of $\mathcal{M}$ in case Simulator exhausts the number of instructions he initially declared would be used to accept or reject. Note that Adam could break Eve's simulation of an accepting run by declaring the value of functions from $\Gamma_{\mathcal{G}_\mathcal{M}}$, which are actually our means of encoding the values of the counters, to be $+\infty$ (or at least some subset of the values of the functions). We describe how we obtain the final set of actions for the constructed game and mention the required transitions from every observation in the game so that Adam is unable to do so—he is able to, but in doing so allows Eve to win the reachability game. Denote by (o, q_i) the i-th state in observation o. Observe that in our construction we need at most 10 states per observation: two copies of every counter state and two additional step counters. $\Sigma = \Sigma' \cup \{q_i \mid 0 \leq i < 6\} \cup \{\mathtt{ex}\}$. For every observation o in $\mathcal{G}_\mathcal{M}$ we add the transitions $((o, q_i), q_i, a^-), ((o, q_j), q_i, a^+)$ for all q_i, q_j in o where $q_i \neq q_j$.

The $\mathtt{ex}$ action is used in transitions $((o, \alpha^+), \mathtt{ex}, \{q_I\}) \in \Delta$ for all observation gadgets o in the universal side of the construction. This allows Eve to stop Adam (who is playing Simulator) in case he tries to simulate more steps than he said were required for $\mathcal{M}$ to reject. Similarly, in the existential part of the construction, we add a transition $((o, \alpha^-), \sigma, \{q_I\})$ for all observation gadgets o and all $\sigma \in \Sigma$, which lets Adam stop Eve's simulation if she tries to cheat in the same way.

To finish, we add the self-loops (a^+, σ, a^+) and (a^-, σ, a^-) as part of Δ for all $\sigma \in \Sigma$. Clearly, Adam cannot choose anything other than proper σ-successors in $\Gamma_{\mathcal{G}_\mathcal{M}}$ or he gives Eve enough information for her to win the game. To have the game be limited observation we let all missing σ-transitions on the existential (resp. universal) side of the simulation go to a sink state in which Adam (Eve) wins.

Correctness. Now, let us prove the correctness of the overall construction. Assume that $\mathcal{M}$ has an accepting or rejecting run. Then, Simulator, by simulating faithfully the run of $\mathcal{M}$ has a strategy that allows him to force abstract paths which induce good or bad abstract cycles depending on who is simulating. Clearly, in this case even if the opponent decides to interrupt the simulation $\mathcal{M}$ at a zero check gadget, he will only be helping Simulator.

If $\mathcal{M}$ has no accepting or rejecting run, then by simulating the machine faithfully, Simulator will be generating cycles in the control state of the machine and such abstract paths are "mixed" because of concrete paths between corresponding α^-, α^+ states. Cheating does not help him either since after the opponent catches him cheating and restarts the simulation of the machine (by returning to the initial observation), the corresponding paths is losing for him. $\square$

Corollary 10 (Class membership)**.** *Let $\mathcal{G}$ be an MPG with limited observation. Determining if $\mathcal{G}$ is forcibly terminating is undecidable.* ∎

Corollary 11 (Winner determination)**.** *Let $\mathcal{G}$ be a forcibly terminating MPG. Determining if Eve wins $\mathcal{G}$ is* R*-complete.* ∎

Proof. R-hardness follows from Theorem 10.4. For decidability, Lemma 10.9 implies that an alternating Turing machine simulating a play on $\Gamma_{\mathcal{G}}$ will terminate. □

We will now give a proof of Theorem 10.1 using a modified version of the above construction.

Proof of Theorem 10.1. We give a reduction from the non-terminating problem for two counter machines using a construction similar to the construction above. Given a two-counter machine $\mathcal{M}$, we construct a game $\mathcal{G}_{\mathcal{M}}$ as in the proof of Theorem 10.4, with the following adjustments:

- We only consider the universal side of the simulation;
- The observation corresponding to the accept state of $\mathcal{M}$ is a sink state winning for Adam;
- The α^- states are replaced with β states which have transitions to other β states of weight 0 except in one case specified below;
- The pumping gadget has self loops of weights $0, 0, -1$ and the transition from b^+ to β has weight -1 if Eve exits and weight 0 if Adam exits;
- The `reset` transition also goes from β states to q_I.

Suppose the counter machine halts in N steps. The strategy for Adam is as follows. Exit the pumping gadget after N steps and faithfully simulate the counter machine. Suppose Eve can beat this strategy. If she allows a faithful simulation for N steps then Adam reaches a sink state and wins, so Eve must play `reset` within N steps of the simulation. Let us consider each cycle of at most $2N$ steps. If she waits for Adam to exit the pumping gadget then the number of steps in the simulation is less than the number of steps in the pumping gadget, so a negative cycle is closed. On the other hand if she exits the pumping gadget before N steps then the cycle through the β vertices has negative weight. In both cases, a negative cycle is closed in at most $2N$ steps, so the limit average is bounded above by $-\frac{1}{2N}$. Thus this strategy is winning for Adam.

Now suppose the counter machine does not halt. The (infinite memory) observation-based strategy for Eve, which we claim is winning for her, is defined as follows. For increasing n, exit the pumping gadget after n steps and faithfully simulate (*i.e.* call any, and only, cheats of Adam) the counter machine for n steps. Then play `reset` and increase n. Cheating in the simulation does not benefit Adam, so we can assume Adam faithfully simulates the counter machine. Likewise, if Eve always waits until the number of steps in the simulation exceeds the number of steps in the pumping gadget, then there is no benefit for Adam to exit the pumping gadget. However if the play proceeds as Eve intends then the weight of the path through the α^+ states is non-negative and although the weight through the β states is negative, the limit average is 0. Thus the strategy is winning for Eve. □

10.3.2 Forcibly first abstract cycle games

In this section and the next we consider restrictions of forcibly terminating games in order to find sub-classes with more efficient algorithmic bounds. The negative algorithmic results from the previous section largely arise from the fact that the abstract cycles required to determine the winner are not necessarily simple cycles. Our first restriction of forcibly terminating games is the restriction of the abstract cycle-forming game to simple cycles.

More precisely, let $\mathcal{G}$ be an MPG with limited observation and $\Gamma_{\mathcal{G}}$ be the associated reachability game. Define $\Pi'_{\mathcal{G}} \subseteq \Pi_{\mathcal{G}}$ as the set of all sequences $x = f_0\sigma_0 f_1\sigma_1 \ldots f_n \in \Pi_{\mathcal{G}}$ such that $\mathsf{supp}(f_i) \neq \mathsf{supp}(f_j)$ for all $0 \leq i < j < n$ and denote by $\Gamma'_{\mathcal{G}}$ the reachability game $(\Pi'_{\mathcal{G}}, \Sigma, f_I, \delta', T'_\exists, T'_\forall)$ where δ' is δ restricted to $\Pi'_{\mathcal{G}}$, $T'_\exists = T_\exists \cap \Pi'_{\mathcal{G}}$ and $T'_\forall = T_\forall \cap \Pi'_{\mathcal{G}}$.

Definition (Forcibly first abstract games)**.** An MPG with limited observation is *forcibly first abstract cycle* (or forcibly FAC) if in the associated reachability game $\Gamma'_{\mathcal{G}}$ either Adam has a winning strategy to reach locations in $T'_\forall$ or Eve has a winning strategy to reach locations in $T'_\exists$. ■

One immediate consequence of the restriction to simple abstract cycles is that the bound in Lemma 10.9 is at most $|\mathsf{Obs}|$. In particular an alternating Turing machine can, in linear time, simulate a play of the reachability game and decide which player, if any, has a winning strategy. Hence the problems of deciding if a given MPG with partial observation is forcibly FAC and deciding the winner of a forcibly FAC game are both solvable in PSPACE. The next results show that there is a matching lower bound for both these problems.

Theorem 10.5 (Class membership)**.** *Let $\mathcal{G}$ be an MPG with limited observation. Determining if $\mathcal{G}$ is forcibly FAC is* PSPACE*-complete.* ■

Proof. For PSPACE membership we observe that a linear bounded alternating Turing machine can decide whether one of the players can force to reach $T'_\exists$ or $T'_\forall$ in $\Gamma_{\mathcal{G}}$. To show hardness we use a reduction from the QBF problem. Let us assume an instance of the problem: a fully quantified Boolean formula $\Psi = \exists x_0 \forall x_1 \ldots \mathcal{Q} x_{n-1}(\Phi)$, where $\mathcal{Q} \in \{\exists, \forall\}$ and Φ is a Boolean formula expressed in *conjunctive normal form* (CNF). We simulate the QBF game (see subsection 2.5.1) with the use of "diamond" gadgets that allow Eve to choose a value for existentially quantified variables by letting her choose the next observation. Similarly, the same gadget—except for the labels on the transitions, which are completely non-deterministic in the following case—allow Adam to choose values for variables that are universally quantified.

We construct a game $\mathcal{G}_\Psi = (Q, q_I, \Sigma, \Delta, \mathsf{Obs}, w)$ in which there are no concrete negative cycles, hence it follows from Lemma 10.4 that there are no bad cycles. The game will thus be forcibly FAC if and only if Eve is able to force good cycles. If Eve is unable to prove the QBF is true, Adam will be able to avoid such plays. For this purpose, the "diamond" gadgets employed have two states per observation. This will allow two disjoint concrete paths to go from the initial state q_I through the whole arena and form a simple abstract cycle that is either good or not good depending on where the cycle started from.

Universally quantified variables. Concretely, let x_1 be a universally quantified variable from Ψ. We add a gadget to $\mathcal{G}_\Psi$ consisting of eight states grouped into four observations: $\{b_0^-, b_0^0\}$, $\{\overline{x_1}, \overline{z_1}\}$, $\{x_1, z_1\}$, $\{b_1^-, b_0^0\}$. We also add the following transitions:

- from b_0^- to $\overline{x_1}$ and x_1, b_0^0 to $\overline{z_1}$ and z_1, with all Σ and weight 0;
- from $\overline{x_1}$ and x_1 to b_1^-, $\overline{z_1}$ and z_1 to b_1^0, with all Σ and the first two with weight -1 while the last two have weight 0.

Figure 10.5 shows the universal "diamond" gadget just described. The observation $\{\overline{x_1}, \overline{z_1}\}$ corresponds to the variable being given a false valuation, whereas the $\{x_1, z_1\}$ observation models a true valuation having been picked. Observe that the choice of the next observation from $\{b_0^-, b_0^0\}$ is completely non-deterministic, *i.e.* Adam chooses the valuation for this variable.

Existentially quantified variables. For existentially quantified variables, the first set of transitions from the gadget is slightly different. Let x_i be an existentially quantified variable in Ψ, then the upper part of the gadget includes transitions from b_i^- to $\overline{x_i}$ and from b_i^0 to $\overline{z_i}$ with action symbol $\neg x_i$ and weight 0; as well as transitions from b_i^- to x_i and from b_i^0 to z_i with action symbol x_i and weight 0.

A play in $\mathcal{G}_\Psi$ traverses gadgets for all the variables from the QBF and eventually gets to the observation $\{b_{n-1}^-, b_{n-1}^0\}$ where the assignment of values for every variable has been simulated. At this point we want to check whether the valuation of the variables makes Φ true. We do so by allowing Adam to choose the next observation (corresponding to one of the clauses from the CNF formula Φ) and letting Eve choose a variable from the clause (which might be negated). Let x_i (resp. $\overline{x_i}$) be the variable chosen by Eve, in $\mathcal{G}_\Psi$ the next observation will correspond to closing a good abstract cycle if and only if the chosen valuation of the variables for Ψ assigns to x_i a true (false) value. For this part of the construction we have $2 \cdot m$ states grouped in m observations, where m is the number of clauses in the formula. The lower part of figure 10.5 shows the clause observations we just described.

Denote by $\{c_i, c_i^0\}$ the observation associated to clause c_i. The game has transitions from c_i to x_i (or $\overline{x_i}$) and from c_i^0 to z_i ($\overline{z_i}$) with action symbol x_i ($\neg x_i$) and weight $n - i$ for the first, 0 for the latter, if and only if the clause c_i includes the (negated) variable x_i.[1]

Correctness. After Eve and Adam have chosen values for all variables (and the game reaches observation $\{b_{n-1}^-, b_{n-1}^0\}$) there are two concrete paths corresponding to the current play: one with payoff 0 and one with payoff $-n$. When Adam has chosen a clause and Eve chooses a variable x_i from the clause, the next observation is reached with both concrete paths having payoffs 0 and $-i$. Observe, however, that if we consider the suffix of said concrete paths starting from $\{x_i, z_i\}$ or $\{\overline{x_i}, \overline{z_i}\}$—depending on which valuation the players chose—both payoffs are 0. Indeed, if the observation was previously visited, *i.e.* Eve has proven the clause to be true, then a good cycle is closed. On the other hand, if the observation has not been visited previously, then Eve has no choice but to keep playing. We note that traversing the lower part of our "diamond" gadgets

[1] All missing transitions for $\mathcal{G}_\Psi$ to be complete go to a dummy state with a negative and 0-valued non-deterministic transitions.

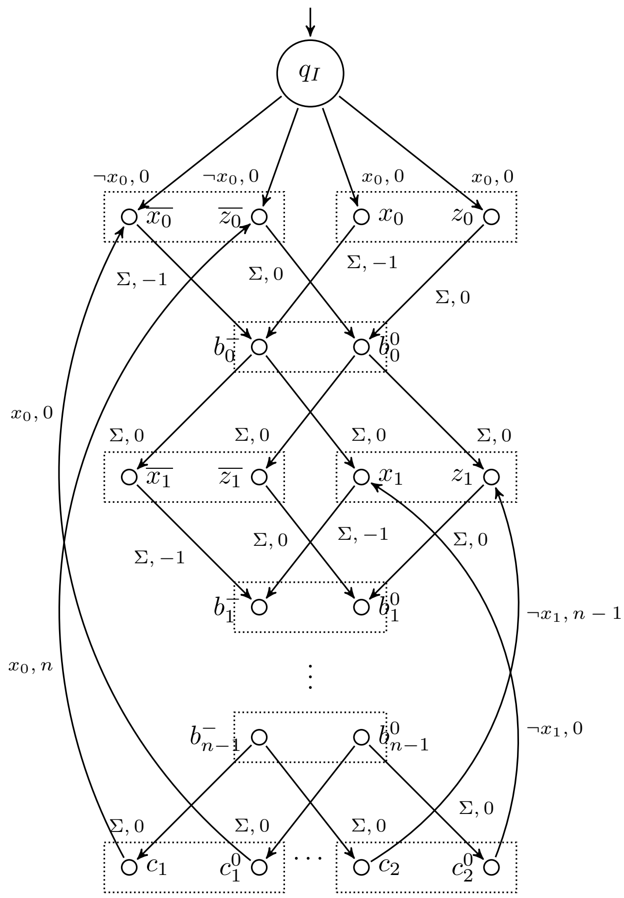

Figure 10.5: Corresponding game for QBF $\exists x_0 \forall x_1 \ldots (\neg x_0) \wedge (x_1) \cdots$

results in a *mixed* payoff of -1 and 0 and since $\{b_i^-, b_i^0\}$ must have already been visited, a cycle is closed that is not good.

Therefore, if Ψ is true then Eve has a strategy to close a good cycle, so $\mathcal{G}_\Psi$ is forcibly FAC. Conversely, if Ψ is false then Adam has a strategy to force Eve to close a cycle that is not good. Hence $\mathcal{G}_\Psi$ is not forcibly FAC. □

We can slightly modify the above construction in such a way that if the game does not finish when the play returns to a variable then Adam can close a bad cycle. This results in a forcibly FAC game that Eve wins if and only if the formula is satisfied. Hence,

Theorem 10.6 (Winner determination)**.** *Let $\mathcal{G}$ be a forcibly FAC MPG. Determining if Eve wins $\mathcal{G}$ is* PSPACE*-complete.* ■

Proof. We describe the modifications required to the construction described above.

First, we augment every observation with $2n$ states corresponding to variables from Φ and their negation (say, y and $\overline{y}$ for $0 \leq i < n$).

We then add transitions from every new state y_i ($\overline{y_i}$) to its counterpart in the next observation so as to form $2n$ new disjoint cycles going from q_I through the whole construction—up to this point $\mathsf{obs}(\mathsf{Plays}(\mathcal{G}_\Psi))$ remains unchanged. These transitions all have weight zero except for a few exceptions:

- the transition corresponding to the lower part of the gadget which represents the variable itself, *i.e.* the transition from augmented observation $\{x_i, z_i, \dots\}$ to $\{b_i^-, b_i^0, \dots\}$ (resp. $\{\overline{x_i}, \overline{z_i}, \dots\}$ to $\{b_i^-, b_i^0, \dots\}$) now has weight of $+1$ for the y_i-transition ($\overline{y_i}$-transition);
- outgoing transitions from clause observations have weight -1 on the y_i-transition going to the x_i-gadget; and
- at every $\{x_i, z_i, \dots\}$ and $\{\overline{x_i}, \overline{z_i}, \dots\}$ augmented observation, Adam is allowed to resolve non-determinism by going back to q_I—*i.e.* in these observations we add a transition from y_i and $\overline{y_i}$, respectively, back to the initial state.

In this new game, we see that when the play reaches $\{x_i, z_i, \dots\}$ (or $\{\overline{x_i}, \overline{z_i}, \dots\}$) after Eve has chosen a variable from a clause then the concrete path ending at y_i (resp. $\overline{y}_i$) has weight 0 if the observation was previously visited, and weight -1 if it was not. The concrete paths ending at all the other new states have weight 0 or $+1$ depending on the choices made by the players. Thus if the observation was previously visited then the cycle closed is good as before, and if the observation was not previously visited then Adam can choose to play to q_I and close a bad cycle. Note that if Adam chooses to play to q_I before the clause gadgets are reached then he will only be closing good cycles. Following the same argument as before, if Ψ is true then Eve has a winning strategy and if Ψ is false then Adam has a winning strategy. So $\mathcal{G}_\Psi$ is forcibly FAC and Eve wins if and only if Ψ is true. □

It also follows from the $|\mathsf{Obs}|$ upper bound on plays in $\Gamma'_\mathcal{G}$ that there is an exponential upper bound on the memory required for a winning strategy for either player. Furthermore, we can show this bound is tight—the games constructed in the proof of Theorem 10.6 can be used to show that there are forcibly FAC games that require exponential memory for winning strategies.

Theorem 10.7 (Exponential memory determinacy)**.** *One player always has a winning observation-based strategy with exponential memory in a forcibly FAC MPG. Further, for any $n \in \mathbb{N}$ there exists a forcibly FAC MPG, of size polynomial in n, such that any winning strategy has memory at least 2^n.* ■

Proof. For the upper bound we observe that plays in $\Gamma'_{\mathcal{G}}$ are bounded in length by $|\mathsf{Obs}|$. It follows that the strategy constructed in Theorem 10.2 has memory at most $|\Sigma|^{|\mathsf{Obs}|}$.

For the lower bound, consider the forcibly FAC game $\mathcal{G}_n$ constructed in the proof of Theorem 10.6 for the formula

$$\varphi_n = \forall x_1 \forall x_2 \ldots \forall x_n \exists y_1 \ldots \exists y_n : \bigwedge_{i=1}^{n} (x_i \vee \neg y_i) \wedge (\neg x_i \vee y_i).$$

As φ_n is satisfied, Eve wins $\mathcal{G}_n$. Now consider any strategy of Eve with memory $< 2^n$. As there are 2^n possible assignments for the values of $x_1, \ldots x_n$ it follows there are at least two different assignments of values such that Eve makes the same choices in the game. Suppose these two assignments differ at x_i and assume w.l.o.g. that Eve's choice is at $(n+i)$-th gadget to play to y_i. Then Adam can win the game by choosing values for the universal variables that correspond to the assignment which sets x_i to false, and then playing to the clause $(x_i, \vee \neg y_i)$. Thus any winning strategy for Eve must have size at least 2^n.

In a similar way the game defined by the formula $\neg\varphi_n$ is won by Adam, but any winning strategy must have size at least 2^n. □

10.3.3 First abstract cycle games

We now consider a structural restriction that guarantees $\Gamma'_{\mathcal{G}}$ is determined.

Definition (First abstract cycle games)**.** An MPG with limited observation is a *first abstract cycle game* (FAC) if in the associated reachability game $\Gamma'_{\mathcal{G}}$ all leaves are in $T'_{\forall} \cup T'_{\exists}$. ■

Intuitively, in an FAC game all simple abstract cycles (that can be formed) are either good or bad. It follows then from Corollary 9 that any cyclic permutation of a good cycle is also good and any cyclic permutation of a bad cycle is also bad. Together with Lemma 10.4, this implies the abstract cycle-forming games associated with FAC games can be seen to satisfy the following three assumptions: (1) A play stops as soon as an abstract cycle is formed; (2) The winning condition and its complement are preserved under cyclic permutations; and (3) The winning condition and its complement are preserved under interleavings. These assumptions correspond to the assumptions required in [AR14] for positional strategies to be sufficient for both players[2]. That is,

Theorem 10.8 (Positional determinacy)**.** *One player always has a positional winning observation-based strategy in an FAC MPG.* ■

As we can check in polynomial time if a positional strategy is winning in an FAC MPG, we immediately have:

[2]These conditions supersede those of [BSV04] which were shown in [AR14] to be insufficient for positional strategies.

Corollary 12 (Winner determination)**.** *Let $\mathcal{G}$ be an FAC MPG. Determining if Eve wins $\mathcal{G}$ is in* NP $\cap$ coNP*.* ■

A path in $\Gamma'_{\mathcal{G}}$ to a leaf not in $T'_{\forall} \cup T'_{\exists}$ provides a short certificate to show that an MPG with limited observation is not FAC. Thus deciding if an MPG is FAC is in coNP. A matching lower bound can be obtained using a reduction from the complement of the Hamiltonian Cycle problem.

Theorem 10.9 (Class membership)**.** *Let $\mathcal{G}$ be an MPG with limited observation. Determining if $\mathcal{G}$ is FAC is* coNP*-complete.* ■

Proof. For coNP membership, one can guess a large enough simple abstract cycle ψ and (in polynomial time with respect to Q) check that it is neither good nor bad. To show coNP-hardness we use a reduction from the complement of the Hamiltonian Cycle problem.

Given graph $\mathcal{G} = (V, E)$ where V is the set of vertices and $E \subseteq V \times V$ the set of edges. We construct a directed weighted graph with limited observation $G = (Q, q_I, \Sigma, \Delta, \mathsf{Obs}, w)$ where:

- $Q = V \cup \{q_I, q_+, q_-\}$;
- $\mathsf{Obs} = \{\{v\} \mid v \in V\} \cup \{\{q_-, q_+\}, \{q_I\}\}$;
- $\Sigma = V \cup \{\tau\}$;
- Δ contains transitions (u, v, v) such that $(u, v) \in E$ and self-loops (u, v', u) for all $(u, v') \notin E$, transitions (with all σ) from q_I to both q_+ and q_- and from these last two to all states $v \in V$, as well as τ-transitions from every state $v \in V$ to q_+ and q_-;
- w is such that all outgoing transitions from q_+ and q_- have weight $1 - |V|$, (u, v, v) transitions where $(u, v) \in E$ have weight $+1$, τ-transitions to q_- from states $v \in V$ have weight -1 and all other transitions have weight 0.

Notice that the only non-deterministic transitions in $\mathcal{G}$ are those incident on and outgoing from the states q_+,q_-. Clearly, the only way for a simple abstract cycle to be not good and not bad (thus making $\mathcal{G}$ not FAC) is if there is a path from $\{q_-, q_+\} \in \mathsf{Obs}$ that traverses $|V|$ unique observations and ends with a τ-transition back at $\{q_-, q_+\}$. Such a path corresponds to a Hamiltonian cycle in $\mathcal{G}$. If there is no Hamiltonian cycle in $\mathcal{G}$ then for any play π in $\mathcal{G}$, a bad cycle will be formed (hence, $\mathcal{G}$ is FAC). □

10.4 Decidable Classes of MPGs with Partial Observation

In the introduction it was mentioned that an MPG with partial observation can be transformed into an MPG with limited observation. The translation from partial observation to limited observation games allows us to extend the notions of FAC and forcibly FAC games to the larger class of MPGs with partial observation. In this section we will investigate the resulting algorithmic effect of this translation on the decision problems we have been considering.

We say an MPG with partial observation is *(forcibly) first belief cycle*, or FBC, if the corresponding MPG with limited observation is (forcibly) FAC.

Our first observation is that FBC MPGs generalize the class of *visible-weights games* studied in [DDG+10]. An MPG with partial observation is considered a visible-weights game if its weight function satisfies the condition that all σ-transitions between any pair of observations have the same weight. We base some of our results for FBC and forcibly FBC games on lower bounds established for problems on visible-weights games.

Lemma 10.10. *Let $\mathcal{G}$ be a visible-weights MPG with partial observation. Then $\mathcal{G}$ is FBC.* ■

We now turn to the decision problems we have been investigating throughout the Chapter. Given the exponential blow-up in the construction of the game of limited observation, it is not surprising that there is a corresponding exponential increase in the complexity of the class membership problem.

Theorem 10.10 (Class membership)**.** *Let $\mathcal{G}$ be an MPG with partial observation. Determining if $\mathcal{G}$ is FBC is* coNEXP*-complete and determining if $\mathcal{G}$ is forcibly FBC is in* EXPSPACE *and* NEXP*-hard.* ■

Proof. Membership of the relevant classes is straightforward, they follow directly from the upper bounds for MPGs with limited observation and the (at worst) exponential blow-up in the translation from games of partial observation to games of limited observation.

Lower bound for FBC games. For coNEXP-hardness we reduce from the complement of the Succinct Hamilton Cycle problem: Given a Boolean circuit C with $2N$ inputs, does the graph on 2^N nodes with edge relation encoded by C have a Hamiltonian cycle? This problem is known to be NEXP-complete [PY86].

The idea is to simulate a traversal of the succinct graph in our MPG: if we make 2^N valid steps without revisiting a vertex of the succinct graph then that guarantees a Hamiltonian cycle. To do this, we start with a transition of weight -2^N and add 1 to all paths every time we make a valid transition. We include a pair of transitions back to the initial state with weights 0 and -1 and ensure this is the only transition that can be taken that results in paths of different weight. The resulting game then has a mixed lasso if and only if we can make 2^N valid transitions. If we encode the succinct graph vertex in the knowledge set then the definition of an FAC game will give us an automatic check if we revisit a vertex. In fact, we store several pieces of information in the knowledge sets of the observations: the current (succinct) graph vertex, the potential successor, and the evaluation of the edge-transition circuit up to a point. We now describe the construction in detail.

Simplifying assumptions. Let us assume inputs of the circuit C are labelled $x_1, \dots x_{2N}$ and that it has k gates $G_1, \dots, G_k$ numbered in an order that respects the circuit graph, so G_j has inputs from $\{x_i, \neg x_i : 1 \leq i < 2N + j\}$ where, for convenience, x_{2N+i} indicates the output of gate G_i. We may assume each gate has two inputs and (as we are allowing negated inputs) we may assume we only have AND and OR gates. The overall (*i.e.* observation-level) structure of the game is shown in Figure 10.6.

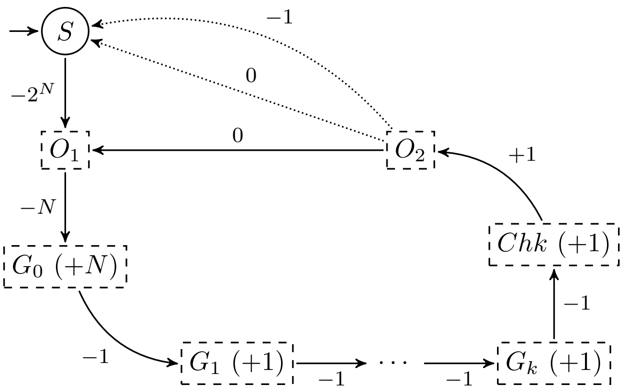

Figure 10.6: Overall structure of the game for Succinct Hamiltonian Cycle

Construction. The game consists of two *external* actions primarily for transitions between observations: σ (solid lines) and σ' (dotted lines); and a number of *internal* actions denoted with τ and χ for transitions primarily within observations (not shown). The numbers in parentheses indicate the maximum weight that can be added to the total with internal transitions, and the edge weights indicate the weight of *all* transitions between observations.

Our game proceeds in several stages:

1. The transition from S to O_1 sets the initial (succinct) vertex (stored in a subset of the states of O_1) and initializes the vertex counter to -2^N.

2. Internal transitions in G_0 select the next vertex, the transition from O_1 to G_0 initializes this procedure.

3. For $i > 0$, internal transitions in G_i evaluate gate i, incoming transitions initialize this by passing on the previous evaluations (including the current and next vertices).

4. Internal transitions in Chk test if the circuit evaluates to 1.

5. The next succinct vertex (chosen in G_0) is passed to O_2, where there is an implicit check that this vertex has not been visited before, and the counter is incremented.

6. The play can return to S, generating a mixed lasso if and only if the vertex counter is 0, *i.e.* 2^N vertices have been correctly visited, or return to O_1 with a new current succinct vertex.

The weights on the incoming transitions to an observation are designed to impose a penalty that can only be nullified if the correct sequence of internal transitions is taken. We observe that if there is a penalty that is not nullified then the game can never enter a mixed lasso (as the vertex counter will still be negative when a vertex is necessarily revisited). We now describe the structure of the observations.

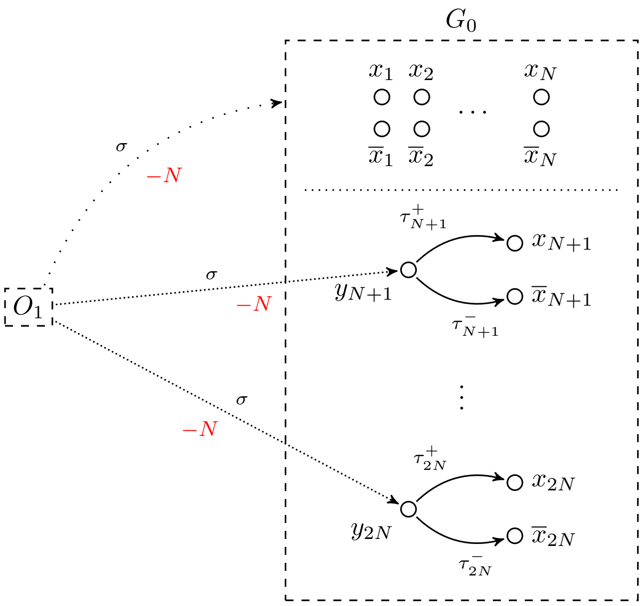

Figure 10.7: Gadget for G_0

(O_1). O_1 contains $2N$ states: $\{x_i, \overline{x}_i \mid 1 \leq i \leq N\}$. For convenience we will use the same labels across different observations, using observation membership to distinguish them. There are σ-transitions from S to $\{x_i \mid 1 \leq i \leq N\}$ with weight -2^N.

(O_2). O_2 contains $2N + 1$ states: $\{x_i, \overline{x}_i \mid 1 \leq i \leq N\} \cup \{\perp\}$. There are σ-transitions from each state in O_2 other than $\perp$ to its corresponding state in O_1 with weight 0. There is a σ'-transition from each state in O_2 other than $\perp$ to S with weight 0, and a σ'-transition from $\perp$ to S with weight -1.

(G_0). G_0 contains $5N$ states: $\{x_i, \overline{x}_i \,|\, 1 \leq i \leq 2N\} \cup \{y_i \,|\, N < i \leq 2N\}$. There is a σ-transition from each state in O_1 to its corresponding state in G_0 of weight $-N$ and in addition, σ-transitions from every state in O_1 to $\{y_i \,|\, N < i \leq 2N\}$ also of weight $-N$. For $N < j \leq 2N$ there is a τ^+_j transition of weight 1 from y_j to x_j and a τ^-_j transition of weight 1 from y_j to $\overline{x}_j$. For all states in G_0 other than y_j there is a τ^+_j and τ^-_j loop of weight 1. Figure 10.7 shows the construction.

G_j ($j > 0$). The observation corresponding to gate j contains $4N + 2j + 8$ states: $\{x_i, \overline{x}_i \mid 1 \leq i \leq 2N + j\} \cup \{v_m, \overline{v}_m \mid 0 \leq m \leq 3\}$. Recall gate j has inputs from $\{x_i, \overline{x}_i \mid 1 \leq i < 2N + j\}$. Suppose these inputs are $y_l \in \{x_l, \overline{x}_l\}$ and $y_r \in \{x_r, \overline{x}_r\}$, and for convenience let $\overline{y}_l$ and $\overline{y}_r$ denote the other member of the pair (*i.e.* the complement of the input). We have a σ-transition of weight -1 from $\{x_i, \overline{x}_i \mid 1 \leq i < 2N + j\} \subseteq G_{j-1}$ to the corresponding state in G_j. In addition we have σ-transitions of weight -1 from $y_l, \overline{y}_l, y_r, \overline{y}_r \in G_{j-1}$ to $v_0, \overline{v}_0, v_1, \overline{v}_1 \in G_j$ respectively. We have the following internal transitions:

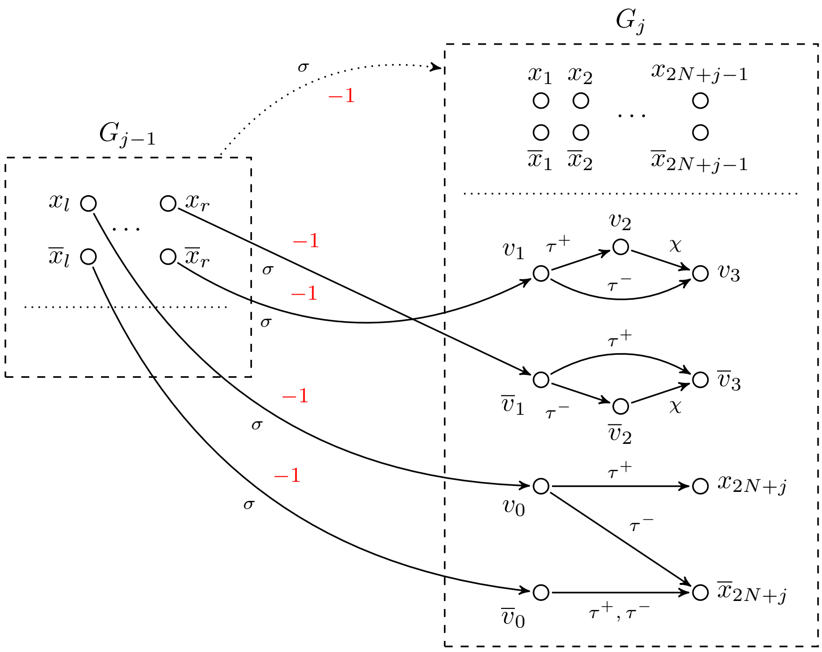

Figure 10.8: Gadget for gate $x_l \wedge \neg x_r$ (self-loops not shown)

- τ^+ (weight 0): v_1 to v_2, $\overline{v}_1$ to $\overline{v}_3$, v_0 to x_{2N+j}, $\overline{v}_0$ to $\overline{x}_{2N+j}$ if gate j is an AND gate, $\overline{v}_0$ to x_{2N+j} if it is an OR gate,
- τ^- (weight 0): v_1 to v_3, $\overline{v}_1$ to $\overline{v}_2$, $\overline{v}_0$ to $\overline{x}_{2N+j}$, v_0 to $\overline{x}_{2N+j}$ if gate j is an AND gate, v_0 to x_{2N+j} if it is an OR gate,
- χ (weight 1): v_2 to v_3, $\overline{v}_2$ to $\overline{v}_3$.

For all other states in G_j these transitions loop with the same weight (*i.e.* χ loops have weight 1, $\tau^{\pm}$ loops have weight 0).

Figure 10.8 shows an example of the construction of G_j for the gate $x_l \wedge \neg x_r$ (self-loops not shown).

(*Chk*). *Chk* contains $4N+2$ states: $\{x_i, \overline{x}_i \mid 1 \leq i \leq 2N\} \cup \{y, z\}$. There is a σ-transition of weight -1 from $\{x_i, \overline{x}_i \mid 1 \leq i \leq 2N\} \subseteq G_k$ to their corresponding states in *Chk*, and a σ-transition of weight -1 from $x_{2N+k} \in G_k$ to y. There is a χ-transition of weight 1 from y to z and for all other states in *Chk* there is a χ-loop of weight 1. There is a σ-transition of weight 1 from all states in *Chk* to $\perp \in O_2$ and for $N < i \leq 2N$ there is a σ-transition of weight 1 from $x_i \in Chk$ to $x_{i-N} \in O_2$ and from $\overline{x}_i \in Chk$ to $\overline{x}_{i-N} \in O_2$.

Lower bound for forcibly FBC games. Suppose we make the following adjustments to the construction:

- Change the weights of incoming transitions to G_i ($i > 0$) to -5 and the weights of all internal τ-transitions to 1,

- Change the weight of the σ'-transition from $\bot \in O_2$ to S to 0,
- Add a new state $\bot$ to all observations other than S (and O_2),
- Add a σ-transition of weight 2^N from S to $\bot \in O_1$, and
- Whenever there is a transition of weight w from $x_i \in o$ to $x_j \in o'$ (o, o' and i, j possibly the same) add a transition of weight $-w$ from $\bot \in o$ to $\bot \in o'$.

Then the only possible non-mixed lasso in the resulting graph[3] is one that would correspond to a successful traversal of a Hamiltonian cycle. Eve can force the play to this cycle if and only if the succinct graph has a Hamiltonian cycle. □

Somewhat surprisingly, for the winner determination problem we have an EXPTIME-time algorithm matching the EXPTIME-hardness lower bound from games with visible weights. This is in contrast to the class membership problem in which an exponential increase in complexity occurs when moving from limited to partial observation.

Theorem 10.11 (Winner determination). *Let $\mathcal{G}$ be a forcibly FBC MPG. Determining if Eve wins $\mathcal{G}$ is* EXPTIME*-complete.* ■

Proof. The lower bound follows from the fact that forcibly FBC games are a generalization of visible-weights games (see Lemma 10.10), shown to be EXPTIME-complete in [DDG+10]. For the upper bound, rather than working on $\Gamma'_{\mathcal{G}'}$, which is doubly-exponential in the size of $\mathcal{G}$, we instead reduce the problem of determining the winner to that of solving a safety game which is only exponential in the size of $\mathcal{G}$. Given an MPG with partial observation $\mathcal{G}$, let $\mathcal{G}'$ be the corresponding limited observation game. Let $\mathcal{E} = \{-1, 0, \ldots, 2w'_{\max} \cdot |\mathsf{Obs}'|\} \cup \{\bot\}$, and let $\mathcal{F}' \subseteq \mathcal{F}$ be the set of functions $f : Q \to \mathcal{E}$.

The safety game will be played on $\mathcal{F}'$ with the transitions defined by σ-successors. The idea is that a given position $f \in \mathcal{F}'$ of the safety game corresponds to being in an observation of $\mathcal{G}'$, namely $\mathsf{supp}(f)$. Similar to before (in $\Gamma_{\mathcal{G}}$), the non-negative integer values of f give a lower bound for the minimum weights of the concrete paths ending in the given state (see Lemma 10.2), that is: if $f(q) \neq \bot$ and $f(q) \geq 0$ then the minimum weight over all concrete paths starting at q_I and ending at q is at least $f(q) + w'_{\max} \cdot |\mathsf{Obs}'|$; whereas if $f(q) = -1$ then there is a concrete path of weight at most $-w'_{\max} \cdot |\mathsf{Obs}'| - 1$. As the winner of a forcibly FAC game can be resolved in at most $|\mathsf{Obs}'|$ transitions it turns out that this is sufficient information to determine the winner.

Formally, the safety game is $\mathcal{S}_{\mathcal{G}} = (\mathcal{F}', f'_I, \Sigma, \Delta_{succ}, \mathcal{F}'_{neg})$ where $f'_I(q_I) = w'_{\max} \cdot |\mathsf{Obs}'|$ and $f'_I(q) = \bot$ for all other $q \in Q$; $(f, \sigma, f') \in \Delta_{succ}$ if f' is a proper σ-successor of f where we let

$$a + b = \begin{cases} \bot & \text{if } a = \bot \text{ or } b = \bot, \\ -1 & \text{if } a = -1,\ b = -1, \text{ or } a + b < 0, \text{ and} \\ \min\{a + b, 2w'_{\max} \cdot |Q|\} & \text{otherwise.} \end{cases}$$

$\mathcal{F}'_{neg}$ is the set of all functions $f \in \mathcal{F}'$ such that $f(q) = -1$ for some $q \in \mathsf{supp}(f)$. The game is played similar to the reachability game $\Gamma_{\mathcal{G}}$, *i.e.* Eve chooses an

[3] We assume dead-ends go to a dummy state with a single mixed self-loop.

action σ and Adam resolves non-determinism by selecting a proper σ-successor. In this case, however, Eve's goal is to avoid visiting any function in $\mathcal{F}'_{neg}$.

The above observation that non-negative values give lower bounds for concrete paths ending at the given state implies that if Eve has a strategy to always avoid $\mathcal{F}'_{neg}$ then $\liminf_{n\to\infty} \frac{\pi[..n]}{n} \geq 0$ for all concrete paths π consistent with the play. That is, if Eve has a winning strategy in $\mathcal{S}_\mathcal{G}$ then she has a winning strategy in $\mathcal{G}$.

Now suppose Eve has a winning strategy in $\mathcal{G}$. It follows from the determinacy of forcibly FAC games and Theorem 10.2 that she has a winning strategy λ in $\Gamma'_{\mathcal{G}'}$. Let λ^* be the translation of λ to $\mathcal{G}'$ as per Theorem 10.2, and let M denote the set of memory states required for λ^*. Clearly λ^* induces a strategy in $\mathcal{S}_\mathcal{G}$. We claim this induced strategy is winning in $\mathcal{S}_\mathcal{G}$. Let $\varrho = f_0\sigma_0\ldots$ be any play in $\mathcal{S}_\mathcal{G}$ consistent with λ^*, and let μ_i denote the i-th memory state obtained in the generation of ϱ (as in Lemma 10.5). Then, with a slight adjustment to the proof of Lemma 10.5 to account for function values not exceeding $2\cdot w_{\max}\cdot|\mathsf{Obs}'|$ we have for all i and all q:

$$\begin{aligned}
f_i(q) - w'_{\max}\cdot|\mathsf{Obs}'| &\geq f_{\mu_i}(q)\\
&= \min\{w(\pi) \mid \pi \in \mathsf{obs}^{-1}(\mathsf{supp}(\mu_i)) \text{ and } \pi \text{ ends at } q\}^4\\
&\geq -w'_{\max}\cdot|\mathsf{Obs}'|
\end{aligned}$$

because $|\mu_i| \leq |\mathsf{Obs}'|$ from the definition of $\Gamma'_{\mathcal{G}'}$. Thus $f_i(q) \geq 0$ for all i. Hence ϱ does not reach $\mathcal{F}'_{neg}$ and is winning for Eve. Thus λ^* is a winning strategy for Eve.

So to determine the winner of $\mathcal{G}$, it suffices to determine the winner of $\mathcal{S}_\mathcal{G}$. This is just the complement of alternating reachability, known to be decidable in polynomial time (see *e.g.* [Pap03]). As

$$|\mathcal{S}_\mathcal{G}| = \mathcal{O}(|\mathcal{F}'|^2) = \mathcal{O}\left((2w'_{\max}\cdot|\mathsf{Obs}'|+1)^{|Q|}\right) = 2^{\mathcal{O}(|Q|^2)},$$

determining the winner of $\mathcal{S}_\mathcal{G}$, and hence $\mathcal{G}$, is in EXPTIME. □

Corollary 13. *Let $\mathcal{G}$ be an FBC MPG. Determining if Eve wins $\mathcal{G}$ is* EXPTIME-*complete.* ■

[4] The step follows from Lemma 10.2.

Chapter 11

Partial-Observation Window Mean-Payoff Games

Window mean payoff (WMP) objectives were recently introduced in [CDRR13] as an alternative to the classical MP objectives. In a WMP objective instead of considering the long-run average along the whole play, payoffs are considered over a local bounded window *sliding* along the play. The objective is then to make sure that the average payoff is at least zero over every window. The WMP objectives enjoy several nice properties. First, in contrast to classical MP objectives, WMP games are decidable even in the partial-observation setting. Second, they can be considered as "approximations" of the classical MP objectives in the following sense: (*i*) they are a *strengthening* of the MP objective, i.e. winning for the WMP objective implies winning for the MP objective, (*ii*) if a (finite memory) strategy guarantees an MP with value $\varepsilon > 0$ then that strategy also achieves the WMP objective for a window size that is bounded by a function of ε, the size of the game, and the memory of the strategy. We remark that, indeed, this is a very weak type of "approximation". However, one cannot hope for much better considering that in [Gen14] it was shown the existence of a polynomial-time approximation scheme for MP objectives would imply that MPGs are solvable in polynomial time.

From a practical point of view, WMP objectives present several advantages. First, they are algorithmically more tractable: in the setting of perfect information games, WMP games can be solved in polynomial-time while the classical MP objectives are only known to be in NP $\cap$ coNP. Second, WMP objectives provide stronger guarantees to the system designer: while classical MP objectives only ensure good performances in the *limit* (long run), variants of WMP objectives provide good performance after a *fixed* or *bounded* amount of time. As we show in this chapter, these advantages transfer to the setting of games with incomplete information, and this is highly desirable for practical purposes. Indeed, to apply synthesis in practice, our models should be as close as possible to the systems that we want to simulate. As classical MPGs with partial-observation leads to undecidability, it is natural to investigate WMP objectives, and in this respect there are two pieces of good news: first, they lead to decidability, and second, there is a potential of algorithmic support with symbolic implementation.

Contributions. In this chapter we consider the extension of WMP objectives to games with partial-observation. We show that, in sharp contrast with classical MP objectives, some of the WMP objectives are decidable for such games. As in [CDRR13], we consider several variants of the window MP objectives. For all objectives, we provide complete complexity results and optimal algorithms. More precisely, our main contributions are as follows:

- First, we consider a definition in which the window size is fixed and the sliding window is started at the initial move of the game, this is called the *direct window* objective. For this definition we give an optimal EXPTIME-time algorithm (Theorem 11.3) in the form of a reduction to a *safety game*. Additionally, we show that this safety game has a nice structure that induces a natural partial order on game positions. In turn this partial order can be used to obtain a *symbolic algorithm* based on the antichain approach [DR10]. This shows that WMP objectives allow us not only to recover decidability but they also lead to games that have the potential to be solved efficiently in practice. The antichain approach has already been applied and implemented with success for LTL synthesis [BBF+12], omega-regular games with partial observation [BCW+09], and language inclusion between non-determinisitic Büchi automata [DR09].
- Second, we consider two natural prefix-independent definitions for the window objectives, the *(uniform) fixed window* objectives. We also give optimal EXPTIME-time algorithms for these two definitions (Theorem 11.5 and Theorem 11.6), when weights are polynomially bounded in the size of the game arena. For these objectives, we show that the sets of good abstract plays (i.e. observation-action sequences) form regular languages whose complements can be recognized by non-deterministic Büchi automata of pseudo-polynomial size (Proposition 11.2 and Proposition 11.3). These automata can then be turned into deterministic parity automata that can be used as observers to transform the game of partial-observation into a game of perfect information with a parity objective.
- Finally, we show that, when the size of the window is not fixed but rather left as a parameter, then for all the objectives that we consider the decision problems are undecidable (Theorem 11.2).

Preliminaries

Given a WA $\mathcal{G}$ and a threshold $\nu \in \mathbb{Q}$, the *mean-payoff* (MP) objectives as

$$\mathsf{MPSup}_{\mathcal{G}}(\nu) = \{\psi \in \mathsf{obs}(\mathsf{Plays}(\mathcal{G})) \mid \forall \pi \in \mathsf{obs}^{-1}(\psi) : \overline{\mathsf{MP}}(\pi) \geq \nu\}$$

and

$$\mathsf{MPInf}_{\mathcal{G}}(\nu) = \{\psi \in \mathsf{obs}(\mathsf{Plays}(\mathcal{G})) \mid \forall \pi \in \mathsf{obs}^{-1}(\psi) : \underline{\mathsf{MP}}(\pi) \geq \nu\}$$

require the mean-payoff value be at least ν. We omit the subscript in the objective names when the WA is clear from the context. Let $\nu = \frac{a}{b}$, w' be a weight function mapping $t \in \Delta$ to $b \cdot w(t) - a$, for all such t, and $\mathcal{G}'$ be the WA resulting from replacing w' in $\mathcal{G}$ for w. We note that Eve wins the $\mathsf{MPSup}_{\mathcal{G}'}(0)$ (respectively, $\mathsf{MPInf}_{\mathcal{G}'}(0)$) objective if and only if she wins $\mathsf{MPSup}_{\mathcal{G}}(\nu)$ (resp., $\mathsf{MPInf}_{\mathcal{G}}(\nu)$).

11.1 Window Mean-Payoff Objectives

In what follows we recall the definitions of the *window mean-payoff* (WMP) objectives introduced in [CDRR13] and adapt them to the partial-observation setting. For the classical MP objectives Eve is required to ensure the long-run average of all concretizations of the play is at least ν. WMP objectives correspond to conditions which are sufficient for this to be the case. All of them use as a main ingredient the concept of concrete paths being "good". Formally, given $i \geq 0$ and *window size bound* $\ell_{\max} \in \mathbb{N}_{>0}$, let the set of concrete paths χ with the *good window* property be

$$\mathsf{GW}(\nu, i, \ell_{\max}) = \{\chi \mid \exists j \leq \ell_{\max} : w(\chi[i..(i+j)]) \geq \nu \cdot j\}.$$

As in [CDRR13], we assume the value of $\ell_{\max}$ is polynomially bounded by the size of the arena.

For the first of the WMP objectives Eve is required to ensure all suffixes of all concretizations of the play can be split into concrete paths of length at most $\ell_{\max}$ and average weight at least ν. The MP objectives are known to be prefix-independent, therefore a prefix-independent version of this first objective is a natural objective to consider as well. We study two such candidates. One which asks of Eve that there is some i such that all suffixes—after i—of all concretizations of the play can be split in the same way as before. This is quite restrictive since the i is *uniform* for all concretizations of the play. The second prefix-independent version of the objective we consider allows for non-uniformity.

Formally, the *fixed window mean-payoff* (FWMP) objectives for a given WA and threshold $\nu \in \mathbb{Q}$ are defined below. For convenience we denote by ψ plays from $\mathsf{obs}(\mathsf{Plays}(\mathcal{G}))$ and concrete plays by π, i.e. elements of $\mathsf{Plays}(\mathcal{G})$.

$$\begin{aligned}
\mathsf{DirFix}(\nu, \ell_{\max}) &= \{\psi \mid \forall \pi \in \mathsf{obs}^{-1}(\psi), \forall i \geq 0 : \pi \in \mathsf{GW}(\nu, i, \ell_{\max})\} \\
\mathsf{UFix}(\nu, \ell_{\max}) &= \{\psi \mid \exists i \geq 0, \forall \pi \in \mathsf{obs}^{-1}(\psi), \forall j \geq i : \pi \in \mathsf{GW}(\nu, j, \ell_{\max})\} \\
\mathsf{Fix}(\nu, \ell_{\max}) &= \{\psi \mid \forall \pi \in \mathsf{obs}^{-1}(\psi), \exists i \geq 0, \forall j \geq i : \pi \in \mathsf{GW}(\nu, j, \ell_{\max})\}
\end{aligned}$$

For the FWMP objectives, we consider $\ell_{\max}$ to be a value that is given as input. Another natural question that arises is whether we can remove this input and consider an even weaker objective in which one asks if there exists an $\ell_{\max}$. This is captured in the definition of the *bounded window mean-payoff* (BWMP) objectives which are defined for a given threshold $\nu \in \mathbb{Q}$.

$$\begin{aligned}
\mathsf{UDirBnd}(\nu) &= \{\psi \mid \exists \ell_{\max}, \forall \pi \in \mathsf{obs}^{-1}(\psi), \forall i \geq 0 : \pi \in \mathsf{GW}(\nu, i, \ell_{\max})\} \\
\mathsf{DirBnd}(\nu) &= \{\psi \mid \forall \pi \in \mathsf{obs}^{-1}(\psi), \exists \ell_{\max}, \forall i \geq 0 : \pi \in \mathsf{GW}(\nu, i, \ell_{\max})\} \\
\mathsf{UBnd}(\nu) &= \{\psi \mid \exists \ell_{\max}, \exists i \geq 0, \forall \pi \in \mathsf{obs}^{-1}(\psi), \forall j \geq i : \pi \in \mathsf{GW}(\nu, j, \ell_{\max})\} \\
\mathsf{Bnd}(\nu) &= \{\psi \mid \forall \pi \in \mathsf{obs}^{-1}(\psi), \exists \ell_{\max}, \exists i \geq 0, \forall j \geq i : \pi \in \mathsf{GW}(\nu, j, \ell_{\max})\}
\end{aligned}$$

As with the mean-payoff objectives we can assume, without loss of generality, that $\nu = 0$. Henceforth, we omit ν.

$$\begin{array}{ccccccccc} \mathsf{DirFix} & & \subset & & \mathsf{UFix} & \subset & \mathsf{Fix} & & \\ \Downarrow & & & & \Downarrow & & \Downarrow & & \\ \mathsf{UDirBnd} & \subset & \mathsf{DirBnd} & \subset & \mathsf{UBnd} & \subset & \mathsf{Bnd} & \subset & \mathsf{MPInf}\ (\subset \mathsf{MPSup}) \end{array}$$

Figure 11.1: Implications among the objectives

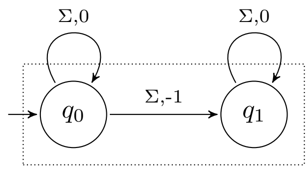

Figure 11.2: Blind WA where, for any $\ell_{\max} \in \mathbb{N}_0$, the only possible abstract play is in $\mathsf{Fix}(\ell_{\max})$ but not in $\mathsf{UFix}(\ell_{\max})$.

11.1.1 Relations among objectives

Figure 11.1 gives an overview of the relative strengths of each of the objectives and how they relate to the mean-payoff objective. The strictness, in general, of most inclusions was established in [CDRR13], and Figure 11.2 provides an example for the remaining case between Fix and UFix.

In general the mean-payoff objective is not sufficient for the FWMP or BWMP objectives, e.g. see Figure 11.3. Our first result shows that if, however, Eve has a *finite memory* winning strategy for a *strictly positive* threshold, then this strategy is also winning for any of the FWMP or BWMP objectives. A specific sub-case of this was first observed in Lemma 2 of [CDRR13].

Theorem 11.1. *Given a WA $\mathcal{G}$, if Eve has a finite memory winning strategy for the $\mathsf{MPInf}(\varepsilon)$ (or $\mathsf{MPSup}(\varepsilon)$) objective, for $\varepsilon > 0$, then the same strategy is winning for her in the $\mathsf{DirFix}_{\mathcal{G}}(\mu)$ game—where μ is bounded by the memory used by the strategy.* ■

Proof. In [DDG+10] the authors show that if Eve is only allowed to play finite memory strategies then she wins the $\mathsf{MPInf}(\nu)$ game if and only if she wins the $\mathsf{MPSup}(\nu)$ game, for any $\nu \in \mathbb{Q}$. We show the claim holds for $\mathsf{MPInf}(\varepsilon)$. Let $\lambda_\exists = (M, m_0, \alpha_u, \alpha_o)$ be the deterministic Mealy machine representation of Eve's finite memory winning strategy. Consider the product of the arena with

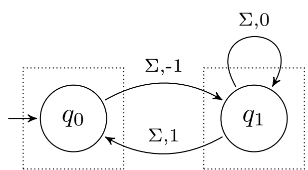

Figure 11.3: Perfect information WA where Eve wins both MP objectives but none of the FWMP or BWMP objectives.

Eve's finite memory winning strategy, $\mathcal{G}\times\lambda_\exists$, constructed in the obvious manner, i.e. every path in $\mathcal{G}\times\lambda_\exists$ corresponds to a concrete path consistent with her strategy. Clearly all cycles in $\mathcal{G}\times\lambda_\exists$ have weight of at least ε, otherwise Adam can create a concrete path with mean-payoff value less than ε by "pumping" the cycles with value less than ε. As any path in $\mathcal{G}\times\lambda_\exists$ corresponds to concrete plays consistent with Eve's strategy, this contradicts the fact that the strategy is winning for her. By Pigeonhole Principle we have that for any path in $\mathcal{G}\times\lambda_\exists$: if a window opens at step i, then after i there is a sequence of length at most $|M||Q|-1$ that is not involved in any cycle. Now, since every cycle has weight $\varepsilon > 0$, after at most

$$\mu = \frac{w_{\max}\cdot|M||Q|}{\varepsilon}\cdot|M||Q|$$

steps the window will have closed. It follows that for all $\psi \in \mathsf{obs}(\mathsf{Plays}(\mathcal{G}))$ consistent with her strategy:

$$\forall\pi\in\mathsf{obs}^{-1}(\psi), \forall i\geq 0 : \pi\in\mathsf{GW}(i,\mu)$$

which concludes our argument. □

11.1.2 Lower bounds

In [CDRR13] it was shown that in multiple dimensions, with arbitrary window size, solving games with the (direct) fixed window objective was complete for EXPTIME-time. We now show that in our setting this hardness result holds, even when the window size is a fixed constant and the weight function is given in unary.

Lemma 11.1. *Let $\ell_{\max}\in\mathbb{N}_{>0}$ be a fixed constant. Given a WA $\mathcal{G}$, determining if Eve has a winning strategy for the $\mathsf{DirFix}(\ell_{\max})$, $\mathsf{UFix}(\ell_{\max})$ or the $\mathsf{Fix}(\ell_{\max})$ objectives is* EXPTIME*-hard, even for unary weights.* ■

Proof. We give a reduction from the problem of determining the winner of a safety game with imperfect information, shown in [CD10] to be EXPTIME-complete.

A safety game with imperfect information is played on a non-weighted game arena with partial-observation $\mathcal{G} = (Q, q_I, \Sigma, \Delta, w, \mathsf{Obs})$. A play of $\mathcal{G}$ is winning for Eve if and only if it never visits the *unsafe state* set $\mathcal{U}\subseteq Q$. Without loss of generality, we assume unsafe states are *trapping*, *i.e.* $(u,\sigma,q)\in\Delta$ and $u\in\mathcal{U}$ imply that $u = q$.

Let w be the transition weight function mapping $(u,\sigma,q)\in\Delta$ to -1 if $u\in\mathcal{U}$ and all other $t\in\Delta$ to 0. Denote by $\mathcal{G}_w$ the resulting WA from adding w to $\mathcal{G}$. It should be clear that Eve wins the safety game $\mathcal{G}$ if and only if she wins $\mathsf{MPInf}_{\mathcal{G}_w}(0)$, $\mathsf{DirFix}_{\mathcal{G}_w}(\ell_{\max})$, $\mathsf{UFix}_{\mathcal{G}_w}(\ell_{\max})$, and $\mathsf{Fix}_{\mathcal{G}_w}(\ell_{\max})$—for any $\ell_{\max}$. That is, all objectives are equivalent for $\mathcal{G}_w$. □

In [CDRR13] the authors show that determining if Eve has a winning strategy in the k-dimensional version of the $\mathsf{UDirBnd}$ and UBnd objectives with perfect information is non-primitive recursive hard. We show that, in our setting, these decision problems are undecidable.

Theorem 11.2. *Given a WA $\mathcal{G}$, determining if Eve has a winning strategy for any of the BWMP objectives is undecidable, even if $\mathcal{G}$ is blind.* ■

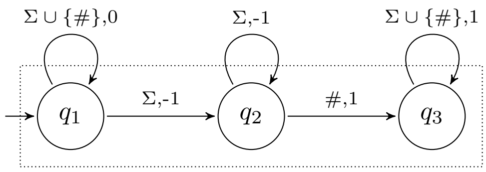

Figure 11.4: Gadget which forces Eve to play infinitely many #.

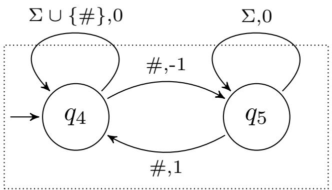

Figure 11.5: Gadget which, given that Eve will play # infinitely often, forces her to play # in intervals of bounded length.

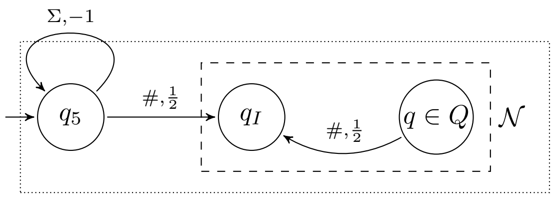

Figure 11.6: Blind gadget to simulate the weighted automaton $\mathcal{N}$.

Proof. We provide a reduction from the universality of weighted finite automata which is undecidable [ABK11]. A *weighted finite automaton* is a tuple $\mathcal{N} = (Q, \Sigma, q_I, \Delta, w)$. A *run* of the automaton on a word $x = \sigma_0\sigma_1 \dots \sigma_n \in \Sigma^*$ is a sequence $r = q_0 q_1 \dots q_n \in Q^+$ such that $(q_i, \sigma_i, q_{i+1}) \in \Delta$ for all $0 \leq i < n$. The cost of the run r is $w(r) = \sum_{i=0}^{n-1} w(q_i, \sigma_i, q_{i+1})$. If the automaton is non-deterministic, it may have several runs on x. In that case, the *cost of x in $\mathcal{N}$* (denoted by $\mathcal{N}(x)$) is defined as the minimum of the costs of all its runs on x.

The *universality* problem for weighted automata is to decide whether, for a given automaton $\mathcal{N}$, the following holds:

$$\forall x \in \Sigma^* : \mathcal{N}(x) < 0.$$

We construct a blind WA, $\mathcal{G}_{\mathcal{N}}$, so that:

- if $\mathcal{N}$ is universal, then Eve has an observation-based winning strategy for the objective $\mathsf{UDirBnd}$,
- if $\mathcal{N}$ is not universal, then Adam has a winning strategy for the complement of the objective Bnd.

As shown in Figure 11.1, $\mathsf{UDirBnd} \subseteq \mathsf{Bnd}$ and all the other BWMP objectives lie in between those two. So, our reduction establishes the undecidability of all BWMP objectives at once.

Our reduction follows the gadgets given in Figures 11.4, 11.5, and 11.6. When the game starts, Adam chooses to play from one of the three gadgets. As the game is blind for Eve, she does not know what is the choice of Adam and so she must be prepared for all possibilities. Note also that as Eve is blind, her strategy can be formalized by an infinite word $w \in \Sigma \cup \{\#\}^\omega$. Let us show first that the two first gadgets force Eve to play a word w such that:

(C_1) there are infinitely many # in w, and

(C_2) there exists a bound $b \in \mathbb{N}$ such that the distance between two consecutive # in w is bounded by b.

Assume that Eve plays a word $w = \#w_1\#w_2\#w_3\#\dots\#w_n\#\dots$ that respects conditions C_1 and C_2, with each $w_i \in \Sigma^*$. First, if Adam decides to play in the first gadget (Figure 11.4), then either Adam stays in state q_1 forever, and he does not open any window, or he decides at some point to go from q_1 to q_2, whereupon he does open a window. However, after at most b steps Adam has to leave q_2 for q_3 at the next occurrence of the # symbol, the bound b is guaranteed by C_2. After at most b additional steps, the open window will be closed as the self loop on q_3 is labelled with the weight $+1$. So in this case, Eve wins the objective $\mathsf{UDirBnd}$. Second, if Adam decides to play in the second gadget (Figure 11.5), then he can go from q_4 to q_5 on the # symbols. The windows that open on those transitions will all close within b steps according to condition C_2 and the game moves back to q_4. So again, Eve wins for the objective $\mathsf{UDirBnd}$.

Now assume that Eve plays a word w that violates either condition C_1 or condition C_2. First, if w violates C_1, then Adam chooses the first gadget (Figure 11.4), and just after Eve has played her last #, Adam moves from q_1 to q_2. As there will be no # anymore, Adam can loop on q_2 and the window that he has opened will never close. Hence, Adam wins for the complement of the objective Bnd. Second, if w violates C_2 then there exists an infinite sequence of indices $i_1 < i_2 < \cdots < i_n < \cdots$ such that $|w_{i_1}| < |w_{i_2}| < \cdots < |w_{i_n}| < \cdots$ Then Adam can read this sequence of sub-words using runs of the form $q_4(q_5)^*q_4$. Each such run will open a window that closes at the end of the sub-word. But as the sequence of lengths of the sub-words is strictly increasing and infinite, Adam wins for the complement of the objective Bnd.

Now, we will assume that Eve plays a word $w = \#w_1\# \dots \#w_n\# \dots$ that respects conditions C_1 and C_2, and we consider what happens when Adam plays in the third gadget (Figure 11.6).

Assume first the automaton $\mathcal{N}$ is *non-universal*. Then by definition, there exists a finite word $w_1 \in \Sigma^*$ such that all runs of $\mathcal{N}$ on w_1 have a non-negative value, i.e. $\mathcal{N}(w_1) \geq 0$. In that case, $w = (\#w_1)^\omega$ is a finite memory winning strategy for Eve for the objective Bnd. Indeed, regardless of which run on w Adam simulates, the mean-payoff of the outcome is at least $\frac{0.5}{b} > 0$ as each new $\#$ brings $+\frac{1}{2}$ and we know that $\mathcal{N}(w_1) \geq 0$. So Eve wins for the objective UDirBnd by Theorem 11.1, as Eve obtains a strictly positive mean-payoff bounded away from zero with a finite memory strategy.

Finally, assume that automaton $\mathcal{N}$ is universal and let us show then that Adam has a winning strategy for the complement of the Bnd objective. Indeed, if Eve plays a word $w = \#w_1\#w_2\#w_3\# \dots \#w_n\# \dots$ that respects conditions C_1 and C_2, then we know that $\mathcal{N}(w_i) < 0$ for each $i \leq 0$. On such word, Adam can follow runs in the gadget of Figure 11.6. As the length between two consecutive $\#$ is at most b, we know that the mean-payoff of the run constructed by Adam is less than or equal to $\frac{-0.5}{b}$. It follows that Adam wins the complement of the Bnd objective as claimed, as Bnd objective implies the mean-payoff objectives (as shown in Figure 11.1). □

11.2 DirFix games

In this section we establish an upper bound to match our lower bound of Section 11.1.2 for determining the winner of DirFix games. We first observe that for WAs with perfect information the DirFix($\ell_{\max}$) objective has the flavor of a safety objective. Intuitively, a play π is winning for Eve if every suffix of π has a prefix of length at most $\ell_{\max}$ with average weight of at least 0. As soon as the play reaches a point for which this does not hold, Eve loses the play. In WAs with partial-observation we need to make sure the former holds for all concretizations of an abstract play.

Consider an abstract path ψ and a positive integer n. We say a window of length l is open at $q \in \mathsf{obs}^{-1}(\psi[n])$ if there is some concretization χ of $\psi[..n]$ with $q = \chi[n]$ such that $\chi \notin \mathsf{GW}(n - l, l)$.

We construct a non-weighted game arena with perfect information $\mathcal{G}'$ from $\mathcal{G}$. Eve's objective in $\mathcal{G}'$ will consist in ensuring the play never reaches locations in which there is an open window of length $\ell_{\max}$, for some state. This corresponds to a safety objective. Whether Eve wins the new game can be determined in time linear w.r.t. the size of the new game (see, e.g. [Tho95]). The game will be played on a set of functions $\mathcal{F}$ which is described in detail below. We then show how to transfer winning strategies of Eve from $\mathcal{G}'$ to $\mathcal{G}$ and vice versa in Lemmas 11.5 and 11.6. Hence, this yields an algorithm to determine if Eve wins the $\mathsf{DirFix}(\ell_{\max})$ objective which runs in exponential-time.

Theorem 11.3. *Given a WA $\mathcal{G}$, determining if Eve has a winning strategy for the $\mathsf{DirFix}(\ell_{\max})$ objective is* EXPTIME*-complete.* ■

Let us define the functions which will be used as the state space of the game. Intuitively, we keep track of the *belief* of Eve as well as the windows with the minimal weight open at every state of the belief.

For the rest of this section let us fix a WA with partial-observation $\mathcal{G}$ and a window size bound $\ell_{\max} \in \mathbb{N}_{>0}$. We begin by defining the set of functions $\mathcal{F}$ as the set of all functions $f : Q \to (\{1, \ldots, \ell_{\max}\} \to \{-w_{\max} \cdot \ell_{\max}, \ldots, 0\}) \cup \{\bot\}$. Denote by $\mathsf{supp}(f)$ the *support* of f, i.e. the set of states $q \in Q$ such that $f(q) \neq \bot$. For $q \in \mathsf{supp}(f)$, we denote by $f(q)_i$ the value $f(q)(i)$. The function $f_I \in \mathcal{F}$ is such that $f_I(q_I)_l = 0$, for all $1 \leq l \leq \ell_{\max}$, and $f_I(q) = \bot$ for all $q \in Q \setminus \{q_I\}$. Given $f_1 \in \mathcal{F}$ and $\sigma \in \Sigma$, we say $f_2 \in \mathcal{F}$ is a σ-successor of f_1 if

- $\mathsf{supp}(f_2) = \mathsf{post}_\sigma(\mathsf{supp}(f_1)) \cap o$ for some $o \in \mathsf{Obs}$;
- for all $q \in \mathsf{supp}(f_2)$ and all $1 \leq j \leq \ell_{\max}$ we have that $f_2(q)_j$ maps to $\max\{-w_{\max} \cdot \ell_{\max}, \min\{0, \zeta(q)\}\}$, where $\zeta(q)$ is defined as follows

$$\zeta(q) = \begin{cases} \displaystyle\min_{\substack{p \in \mathsf{supp}(f_1) \land (p,\sigma,q) \in \Delta, \\ f_1(p)_{j-1} < 0}} f_1(p)_{j-1} + w(p, \sigma, q) & \text{if } j \geq 2 \\ \displaystyle\min_{p \in \mathsf{supp}(f_1) \land (p,\sigma,q) \in \Delta} w(p, \sigma, q) & \text{otherwise.} \end{cases}$$

Lemma 11.2. *The number of elements in $\mathcal{F}$ is at most $2^{|Q| \cdot \ell_{\max} \cdot \log(w_{\max} \cdot \ell_{\max})}$.* ■

Proof.

$$\begin{aligned} |\mathcal{F}| &\leq (w_{\max} \cdot \ell_{\max})^{|Q| \cdot \ell_{\max}} \\ &= \left(2^{\log(w_{\max} \cdot \ell_{\max})}\right)^{|Q| \cdot \ell_{\max}} \\ &= 2^{|Q| \cdot \ell_{\max} \cdot \log(w_{\max} \cdot \ell_{\max})}. \end{aligned}$$

Hence, the result holds. □

We extend the supp operator to finite sequences of functions and actions. In other words, given $\varrho' = f_0 \sigma_0 f_1 \sigma_1 \in (\mathcal{F} \cdot \Sigma)^*$, $\mathsf{supp}(\varrho') = s_0 \sigma_0 s_1 \sigma_1 \ldots$ where $s_i = \mathsf{supp}(f_i)$ for all $i \geq 0$. In an abuse of notation, we define the function $\mathsf{supp}^{-1} : (\mathsf{Obs} \cdot \Sigma)^* \times \mathcal{F} \to (\mathcal{F} \cdot \Sigma)^*$ which maps abstract paths to function-action sequences. Formally, given $\varrho = o_0 \sigma_0 o_1 \sigma_1 \cdots \in \mathsf{obs}(\mathsf{Prefs}(())\mathcal{G})$ and $\varphi \in \mathcal{F}$ with $\mathsf{supp}(\varphi) \subseteq o_0$, $\mathsf{supp}^{-1}(\varrho, \varphi) = f_0 \sigma_0 f_1 \sigma_1 \ldots$ where $f_0 = \varphi$ and for all $i \geq 0$ we have that f_{i+1} is the σ_i-successor of f_i such that $\mathsf{supp}(f_{i+1}) \subseteq o_{i+1}$. Both supp and supp^{-1} are extended to infinite sequences in the obvious manner.

The following two results enunciate the key properties of sequences of the form $(\mathcal{F} \cdot \Sigma)^*$. Intuitively, the set of all those sequences corresponds to the windowed, weighted unfolding of $\mathcal{G}$ with information about reachable states as well as open windows.

Lemma 11.3. *Let $\varrho = o_0\sigma_0 \dots o_n$ be an abstract path, $\varphi \in \mathcal{F}$ such that $\mathsf{supp}(\varphi) \subseteq o_0$ and $\mathsf{supp}^{-1}(\varrho, \varphi) = f_0\sigma_0 \dots f_n \in (\mathcal{F} \cdot \Sigma)^*$. A state $q \in Q$ is reachable from some state $q_0 \in \mathsf{supp}(\varphi)$ through a concrete path $q_0\sigma_0 \dots q_n \in \mathsf{obs}^{-1}(\varrho)$ if and only if $q \in \mathsf{supp}(f_n)$.* ■

Proof. ($\Rightarrow$) We proceed by induction. We will show that for all $0 \leq j \leq n$, for all $q_j \in \mathsf{supp}(f_j)$ there is a concrete path $q_0\sigma_0 \dots q_j$ such that $q_k \in o_k$ for all $1 \leq k \leq j$ and $q_0 \in \mathsf{supp}(\varphi)$. Note that for $j = 0$ the claim trivially holds. Assume the claim holds for j. From the definition of σ-successor and supp^{-1} we have that $\mathsf{supp}(f_{j+1}) = \mathsf{post}_{\sigma_j}(\mathsf{supp}(f_j)) \subseteq o_{j+1}$. This means that for all $q_{j+1} \in \mathsf{supp}(f_{j+1})$ there must be some $q_j \in \mathsf{supp}(f_j)$ such that $(q_j, \sigma_j, q_{j+1}) \in \Delta$. Hence any q_{j+1} is reachable from some q_j via σ_j which, by inductive hypothesis, is in turn reachable from some $q_0 \in \mathsf{supp}(\varphi)$ via a concrete path of the desired form.

($\Leftarrow$) We now show—once more by induction on j—that for all $0 \leq j \leq n$, if there is a concrete path $q_0\sigma_0 \dots q_j$ such that $q_0 \in \mathsf{supp}(\varphi)$ and $q_k \in o_k$ for all $1 \leq k \leq j$, then $q_j \in \mathsf{supp}(f_j)$. The claim holds for $j = 0$. Assume that it holds for some j. From the assumptions we have that $(q_j, \sigma_j, q_{j+1}) \in \Delta$ and $q_{j+1} \in o_{k+1}$. Further, we know that $q_j \in \mathsf{supp}(f_j)$ by inductive hypothesis. Hence, $q_{j+1} \in \mathsf{post}_{\sigma_j}(\mathsf{supp}(f_j)) \subseteq o_{j+1}$ which means that $q_{j+1} \in \mathsf{supp}(f_{j+1})$. □

Lemma 11.4. *Let $\varrho = o_0\sigma_0 \dots o_n$ be an abstract path, $\varphi \in \mathcal{F}$ such that $\mathsf{supp}(\varphi) \subseteq o_0$ and $\mathsf{supp}^{-1}(\varrho, \varphi) = f_0\sigma_0 \dots f_n \in (\mathcal{F} \cdot \Sigma)^*$. Given state $p \in \mathsf{supp}(f_n)$ and $1 \leq l \leq \ell_{\max}$ such that $l \leq n$, then there is a window of length l open at p if and only if $f_n(p)_l < 0$.* ■

Proof. Instead of directly providing a proof of Lemma 11.4, we prove a more general result below. Consider the three conditions stated in Claim 15. We shall prove that C1 $\Rightarrow$ C2 $\Rightarrow$ C3 $\Rightarrow$ C1. Since C1 corresponds to having a window of length l open at p from $\mathsf{supp}(\varphi)$, the desired result follows from transitivity.

Claim 15. *Let $\varrho = o_0\sigma_0 \dots o_n$ be an abstract path, $\varphi \in \mathcal{F}$ such that $\mathsf{supp}(\varphi) \subseteq o_0$ and $\mathsf{supp}^{-1}(\varrho, \varphi) = f_0\sigma_0 \dots f_n \in (\mathcal{F} \cdot \Sigma)^*$. Given state $p \in \mathsf{supp}(f_n)$ and $1 \leq l \leq \ell_{\max}$ such that $l \leq n$, let $\lambda = n - l$. The following three statements are equivalent.*

C1. *There is a concrete path $q_0\sigma_0 \dots q_n \in \mathsf{obs}^{-1}(\varrho)$ with $q_n = p$ and $q_0 \in \mathsf{supp}(\varphi)$ and*
$$\sum_{j=n-l}^{m} w(q_j, \sigma_j, q_{j+1}) < 0$$
for all $n - l \leq m < n$.

C2. $f_n(p)_l < 0$.

C3. *There is a concrete path $q_0\sigma_0 \dots q_n \in \mathsf{obs}^{-1}(\varrho)$ with $q_n = p$ and $q_0 \in \mathsf{supp}(\varphi)$ such that*

(a) $f_j(q_j)_{j-\lambda} < 0$ *for all* $\lambda < j \leq n$, *and*

(b) $f_k(q_k)_{j-\lambda} + w(q_k, \sigma_k, q_{k+1}) = f_{k+1}(q_{k+1})_{k-\lambda+1}$ *for all* $\lambda < k < n$.

■

(C3 ⇒ C1). We will apply induction on m. From the definition of σ-successor we have that $f_{\lambda+1}(q_{\lambda+1})_1 = \min\{0, w(q_\lambda, \sigma_\lambda, q_{\lambda+1})\}$. From assumption (a) we know that $f_{\lambda+1}(q_{\lambda+1})_1 < 0$. Thus, the claim holds for $m = \lambda$. Assume it holds for m. To conclude the proof, we now show that the claim holds for $m+1$ as well.

$$\begin{aligned}\sum_{j=n-l}^{m+1} w(q_j, \sigma_j, q_{j+1}) &= f_m(q_m)_{m-\lambda} + w(q_m, \sigma_m, q_{m+1}) && \text{ind. hyp.}\\ &= f_m(q_m)_{m-\lambda+1} && \text{from } (b)\\ &< 0 && \text{from } (a).\end{aligned}$$

(C1 ⇒ C2). We show, by induction on m, that for all $\lambda \leq m < n$

$$f_{m+1}(q_{m_1})_{m-\lambda+1} \leq \sum_{j=\lambda}^{m} w(q_j, \sigma_j, q_{j+1}).$$

The desired result follows. As the base case, consider $m = \lambda$ and note that by definition of σ-successor we have that

$$\begin{aligned}f_{\lambda+1}(q_{\lambda+1})_1 &= \min(\{0\} \cup \{w(p, \sigma_\lambda, q_{\lambda+1}) \mid p \in \mathsf{supp}(f_\lambda) \wedge (p, \sigma_\lambda, q_{\lambda+1}) \in \Delta\})\\ &\leq w(q_\lambda, \sigma_\lambda, q_{\lambda+1}).\end{aligned}$$

Thus the claim holds. Assume that the claim is true for m. From the definition of σ-successor we have that

$$f_{m+2}(q_{m+2})_{\lambda-m+2} \leq f_{m+1}(q_{m+1})_{m-\lambda+1} + w(q_{m+1}, \sigma_{m+1}, q_{m+2}).$$

From the inductive hypothesis we get have that the right hand side of the inequality is equivalent to

$$\sum_{j=\lambda}^{m+1} w(q_j, \sigma_j, q_{j+1}).$$

Thus the claim holds for $m+1$ as well.

(C2 ⇒ C3). We inductively construct a concrete path $q_0\sigma_0 \dots q_n\mathsf{obs}^{-1}(\varrho)$ with $q_n = p$ and $q_0 \in \mathsf{supp}(\varphi)$ such that

(1) $f_{n-k}(q_{n-k})_{l-k} < 0$ for all $0 \leq k < l$, and

(2) $f_{n-k}(q_{n-k})_{l-k} = w(q_{n-k+1}, \sigma_{n-k+1}, q_{n-k}) + f_{n-k+1}(q_{n-k+1})_{l-k+1}$ for all $1 \leq k < l$.

As these conditions are equivalent to (a)-(b) from C3, the result follows. Note that for $k = 0$ we have that (1) holds trivially since $p \in \mathsf{supp}(f_n)$ and $f_n(p)_l < 0$ by hypothesis. If $f_{n-k}(q_{n-k})_{l-k} < 0$ then, by definition of σ-successor, it follows that there is some $q' \in \mathsf{supp}(f_{n-k+1}) \subseteq o_{n-k+1}$ such that $f_{n-k+1}(q')_{l-k+1} < 0$ and $f_{n-k}(q_{n-k})_{l-k} = w(q_{n-k+1}, \sigma_{n-k+1}, q_{n-k}) + f_{n-k+1}(q_{n-k+1})_{l-k+1}$. In other words, q' is the source of the minimal σ_{n-k+1}-transition of a state from $\mathsf{supp}(f_{n-k+1})$ to q_{n-k}. Let $q_{n-k+1} = q'$. Continue in this fashion defining every q_i up to q_{n-l}. Now, from Lemma 11.3, we have that q_{n-l} is reachable from some state in $\mathsf{supp}(\varphi)$ via a concrete path of the desired form. Any such path is a valid prefix for the sequence $q_{n-l}\sigma_{n-l} \dots q_n$ we constructed above. □

Formally, the arena $\mathcal{G}' = (\mathcal{F}, f_I, \Sigma, \Delta')$. The transition relation Δ' contains the transition (f_1, σ, f_2) if f_2 is the σ-successor of f_1. Eve, in $\mathcal{G}'$, is required to avoid states $\mathcal{U} = \{f \in \mathcal{F} \mid \exists q \in \mathsf{supp}(f) : f(q)_{\ell_{\max}} < 0\}$.

Lemma 11.5. *If Eve wins the safety objective in $\mathcal{G}'$, then she also wins the $\mathsf{DirFix}(\ell_{\max})$ objective in $\mathcal{G}$.* ■

Proof. Assume λ' is a winning strategy for Eve in $\mathcal{G}'$. We define a strategy λ for her in $\mathcal{G}$ as follows: $\lambda(\varrho) = \lambda'(\mathsf{supp}^{-1}(\varrho, f_I))$ for all $\varrho \in \mathsf{obs}(\mathsf{Prefs}(())\mathcal{G})$. We claim that λ is winning for her in $\mathcal{G}$. Towards a contradiction, assume $\psi \in \mathsf{obs}(\mathsf{Plays}(\mathcal{G}))$ is consistent with λ and that $\psi \notin \mathsf{DirFix}(\ell_{\max})$. Recall that this implies there are $n \in \mathbb{N}, q \in Q$ such that there is a window of length $\ell_{\max}$ open at $q \in \mathsf{obs}^{-1}(\psi[n])$. By Lemma 11.4 we then get that f_n from $\mathsf{supp}^{-1}(\psi, f_I) = f_0\sigma_0 f_1\sigma_1 \dots$ is in $\mathcal{U}$. As $\mathsf{supp}^{-1}(\psi, f_I)$ is consistent with λ', this contradicts the assumption that λ' was winning. □

Lemma 11.6. *If Eve wins the $\mathsf{DirFix}(\ell_{\max})$ objective in $\mathcal{G}$, then she also wins the safety objective in $\mathcal{G}'$.* ■

Proof. Assume λ is a winning strategy for Eve in $\mathcal{G}$. We define a strategy λ' for her in $\mathcal{G}'$ as follows: $\lambda'(\varrho') = \lambda \circ \mathsf{obs} \circ \mathsf{supp}(\varrho')$ for all $\varrho' \in \mathsf{obs}(\mathsf{Prefs}(())\mathcal{G}')$. We claim that λ' is winning for her in $\mathcal{G}'$. Again, towards a contradiction, assume $\psi' \in \mathsf{obs}(\mathsf{Plays}(())\mathcal{G}')$ is consistent with λ' and that ψ' visits some $f \in \mathcal{U}$. This implies, by Lemma 11.4, that there is a window of length $\ell_{\max}$ open at some $q \in \mathsf{supp}(f)$ in $\psi = \mathsf{obs}(\mathsf{supp}(\psi'))$. As Δ is total, for any $\sigma \in \Sigma$ Eve plays then there is valid σ-successor of q that Adam can choose as the next state. Hence there is some $\chi \in \mathsf{obs}(\mathsf{Plays}(\mathcal{G}))$ consistent with λ such that χ and ψ have the same prefix up to i_q, where $q \in \mathsf{obs}^{-1}(\chi[i_q])$, and there is a concretization π of χ such that $\pi[i_q] = q$. As χ is consistent with λ and $\chi \notin \mathsf{DirFix}(\ell_{\max})$, this contradicts the fact that it was a winning strategy. □

11.2.1 A symbolic algorithm for DirFix games

We note that the state space of the construction $\mathcal{G}'$ presented in Section 11.2 admits an order such that if a state is smaller than another state, according to said order, and Eve has a strategy to win from the latter, then she has a strategy to win from the former. In this section we formalize this notion by defining the order and, in line with [CDHR06, BBF+12], propose an *antichain*-based algorithm to solve the safety game on $\mathcal{G}'$.

We define the *uncontrollable predecessors* operator $\mathsf{UPre} : \mathcal{P}(\mathcal{F}) \to \mathcal{P}(\mathcal{F})$ as

$$\mathsf{UPre}(S) = \{p' \in \mathcal{F} \mid \forall \sigma \in \Sigma, \exists q' \in S : (p', \sigma, q') \in \Delta'\}.$$

For $S \in \mathcal{P}(\mathcal{F})$, we denote by $\mu X.(S \cup \mathsf{UPre}(X))$, the *least fixpoint* of the function $F : X \to S \cup \mathsf{UPre}(X)$ in the μ-calculus notation (see [EJ91]). Note that F is defined on the power-set lattice, which is finite. The following is a well-known result about the relationship between safety games and the UPre operator (see e.g. [Grä04]).

Proposition 11.1. *Eve wins a safety game with unsafe state set $\mathcal{U}$ if and only if the initial state of the game is not contained in $\mu X.(\mathcal{U} \cup \mathsf{UPre}(X))$.* ■

Definition (The partial order)**.** Given $f', g' \in \mathcal{F}$ we say $f' \preceq g'$ if and only if $\mathsf{supp}(f') \subseteq \mathsf{supp}(g')$ and

$$\forall q \in \mathsf{supp}(f'), \forall i \in \{1, \ldots, \ell_{\max}\}, \exists j \in \{i, \ldots, \ell_{\max}\} : f'(q)_i \geq g'(q)_j.$$

■

An *antichain* is a non-empty set $S \in \mathcal{P}(\mathcal{F})$ such that for all $x, y \in S$ we have $x \not\preceq y$. We denote by $\mathfrak{A}$ the set of all antichains. Given $a, b \in \mathfrak{A}$, denote by $a \sqsubseteq b$ the fact that $\forall x \in b, \exists y \in a : y \preceq x$. For $S \in \mathcal{P}(\mathcal{F})$ we denote by $\lfloor S \rfloor$ the set of minimal elements of S, that is $\lfloor S \rfloor = \{x \in S \mid \forall y \in S : y \preceq x$ implies $y = x\}$. Clearly $\lfloor S \rfloor$ is an antichain.

Given $S \in \mathcal{P}(\mathcal{F})$ we denote by $S\!\uparrow$ the *upward-closure* of S, that is $S\!\uparrow = \{t \in \mathcal{F} \mid S \preceq t\}$. We say a set $s \in \mathcal{P}(\mathcal{F})$ is *upward-closed* if $S = S\!\uparrow$. Note that $\lfloor S \rfloor\!\uparrow = S\!\uparrow$ and therefore, if S is upward-closed, the antichain $\lfloor S \rfloor$ is a succinct representation of S.

Lemma 11.7. *The following assertions hold.*

1. *$\mathcal{U}$ is upward-closed.*
2. *If $S, T \in \mathcal{P}(\mathcal{F})$ are two upward-closed sets, then $S \cup T$ is also upward-closed.*

■

The usual way of showing an antichain algorithm works dictates that we now prove the UPre operator, when applied to upward-closed sets, outputs an upward-closed set as well. Unfortunately, this is not true in our case. The following example illustrates this difficulty.

Example 7. Consider the WA from Figure 11.3 and let $\ell_{\max} = 2$. We note that the function f such that $f(q_0) = \bot$ and $f(q_1)_1 = 1, f(q_1)_2 = 0$ is in $\mathsf{UPre}(\mathcal{U})$. We also have that for the function g such that $g(q_0) = \bot$ and $g(q_1)_1 = 0, g(q_1)_2 = 1$ we get that $f \preceq g$. It is easy to verify $g \notin \mathsf{UPre}(\mathcal{U})$. Hence, $\mathsf{UPre}(\mathcal{U})$ is not upward-closed. ■

However, we claim that one can circumvent this issue by ignoring elements from $\mathcal{U}$. Thus we are able to prove that, under some conditions, UPre does preserve "upward-closedness".

Lemma 11.8. *Given upward-closed set $S \in \mathcal{P}(\mathcal{F})$ and $f, g \in \mathcal{F} \setminus \mathcal{U}$, if $f \in \mathsf{UPre}(S)$ and $f \preceq g$, then $g \in \mathsf{UPre}(S)$.* ■

Proof. We have that for all σ, there is $h_\sigma \in S$ such that $(f, \sigma, h_\sigma) \in \Delta'$. By construction of Δ' we also know that there is i_σ such that $(g, \sigma, i_\sigma) \in \Delta'$, and furthermore, since $\mathsf{supp}(f) \subseteq \mathsf{supp}(g)$, we get that

$$\begin{aligned}
\mathsf{supp}(h_\sigma) &= \mathsf{post}_\sigma(\mathsf{supp}(f)) \cap o \\
&\subseteq \mathsf{post}_\sigma(\mathsf{supp}(g)) \cap o \\
&= \mathsf{supp}(i_\sigma)
\end{aligned}$$

for some $o \in \mathsf{Obs}$. Note that:

(1) since $f, g \notin \mathcal{U}$, then $f(p)_{\ell_{\max}} = g(p)_{\ell_{\max}} = 0$ for all $p \in \mathsf{supp}(f)$; and

(2) $i_\sigma(q)_1 = h_\sigma(q)_1$ for all $q \in \mathsf{supp}(h_\sigma)$.

From (1) and since $f \preceq g$, there is a function $\alpha : \{1, \ldots, \ell_{\max}\} \to \{1, \ldots, \ell_{\max} - 1\}$ such that for all $1 \leq x < \ell_{\max}$ we have that $\alpha(x) \geq x$ and $f(p)_x \geq g(p)_{\alpha(x)}$ holds for all $p \in \mathsf{supp}(f)$. Observe that for all $q \in \mathsf{supp}(h_\sigma)$ and any $2 \leq x \leq \ell_{\max}$, we have that

$$\begin{aligned} h_\sigma(q)_x &= \min_{p \in \mathsf{supp}(f)} (\{0\} \cup \{f(p)_{x-1} + w(p, \sigma, q) \mid f(p)_{x-1} < 0\} \\ &\geq \min_{p \in \mathsf{supp}(f)} (\{0\} \cup \{g(p)_{\alpha(x-1)} + w(p, \sigma, q) \mid g(p)_{\alpha(x-1)} < 0\} \\ &\geq i_\sigma(q)_{\alpha(x-1)+1}. \end{aligned}$$

It follows that $h_\sigma \preceq i_\sigma$ and that, since S is upward-closed, $i_\sigma \in S$. Thus, we have shown that for all σ, there is $i_\sigma \in S$ such that $(g, \sigma, i_\sigma) \in \Delta'$, which implies that $g \in \mathsf{UPre}(S)$. □

We define a version of the uncontrollable predecessors' operator which manipulates antichains instead of subsets of $\mathcal{F}$.

$$\lfloor \mathsf{UPre} \rfloor(a) = \lfloor \{p' \in \mathcal{F} \setminus \mathcal{U} \mid \forall \sigma \in \Sigma, \exists q' \in a, \exists r' \in \mathcal{F} : (p', \sigma, r') \in \Delta' \wedge q' \preceq r'\} \rfloor$$

Given $a, b \in \mathfrak{A}$ we denote by $a \sqcup b$ the *least upper bound* of a and b, i.e. $a \sqcup b = \lfloor \{q' \in \mathcal{F} \mid q' \in a \text{ or } q' \in b\} \rfloor$. It is easy to check that $(a \sqcup b)\uparrow = a\uparrow \cup b\uparrow$ for any $a, b \in \mathfrak{A}$.

Theorem 11.4. *Given a WA $\mathcal{G}$, Eve wins the $\mathit{DirFix}(\ell_{\max})$ objective if and only if $\{q_I'\} \not\sqsupseteq \mu X.(\lfloor \mathcal{U} \rfloor \sqcup \lfloor \mathsf{UPre} \rfloor(X))$.* ■

Before proving the above theorem, we first argue the following holds.

Lemma 11.9. *Given upward-closed set $S \in \mathcal{P}(\mathcal{F})$, $\lfloor \mathsf{UPre} \rfloor(\lfloor S \rfloor) = \lfloor \mathsf{UPre}(S) \setminus \mathcal{U} \rfloor$.* ■

Proof. We first show that if $f \in \lfloor \mathsf{UPre} \rfloor(\lfloor S \rfloor)$ then $f \in \mathsf{UPre}(S) \setminus \mathcal{U}$. We have that $f \notin \mathcal{U}$ and $\forall \sigma \in \Sigma, \exists q' \in \lfloor S \rfloor, \exists r'_\sigma \in \mathcal{F} : (f, \sigma, r'_\sigma) \in \Delta'$ and $q' \preceq r'_\sigma$. Since S is upward-closed and $q' \preceq r'_\sigma$, we know that $r'_\sigma \in S$. Hence, we get that $\forall \sigma \in \Sigma, \exists r'_\sigma \in S : (f, \sigma, r'_\sigma) \in \Delta'$, which implies that $f \in \mathsf{UPre}(S) \setminus \mathcal{U}$.

Next, we show that if $f \in \lfloor \mathsf{UPre}(S) \setminus \mathcal{U} \rfloor$ then $f \in \{p' \in \mathcal{F} \setminus \mathcal{U} \mid \forall \sigma \in \Sigma, \exists q' \in a, \exists r' \in \mathcal{F} : (p', \sigma, r') \in \Delta'$ and $q' \preceq r'\}$. We know that $f \notin \mathcal{U}$ and $\forall \sigma \in \Sigma, \exists r' \in Q : (f, \sigma, r') \in \Delta'$. By definition of $\lfloor S \rfloor$, we know there is $q_{r'} \in \lfloor S \rfloor$ such that $q_{r'} \preceq r'$. Thus, we get that $\forall \sigma \in \Sigma, \exists r'_\sigma \in S, \exists q_{r'} \in \lfloor S \rfloor : (f, \sigma, r') \in \Delta'$ and $q_{r'} \preceq r'$.

Finally, we note that if $f \in \lfloor \mathsf{UPre} \rfloor(\lfloor S \rfloor)$ then not only is it true that $f \in \mathsf{UPre}(S) \setminus \mathcal{U}$, but furthermore $f \in \lfloor \mathsf{UPre}(S) \setminus \mathcal{U} \rfloor$. Indeed, if this were not the case, then there would be $g \in \lfloor \mathsf{UPre}(S) \setminus \mathcal{U} \rfloor$ such that $g \preceq f$ and $f \neq g$. Then, by the argument explained in the previous paragraph, this would contradict minimality of f in $\lfloor \mathsf{UPre} \rfloor(\lfloor S \rfloor)$. Similarly, if $f \in \lfloor \mathsf{UPre}(S) \setminus \mathcal{U} \rfloor$ then $f \in \lfloor \mathsf{UPre} \rfloor(\lfloor S \rfloor)$, as otherwise, by the argument from the first paragraph of the proof, minimality in the first set would be contradicted. Thus, the claim holds. □

We are now ready to present our proof for the theorem.

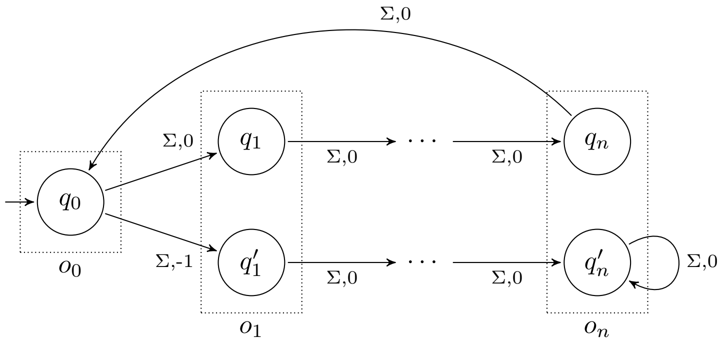

Figure 11.7: For $n > \ell_{\max} + 1$ the abstract path $(o_0 \dots o_n)^\omega$ is winning for the Fix condition but infinitely often visits an unsafe state in the construction from Section 11.2.

of Theorem 11.4. We note that for any upward-closed set $S \subseteq \mathcal{F}$ such that $\mathcal{U} \subseteq S$ we have, from Lemma 11.8 that $\mathcal{U} \cup \mathsf{UPre}(S)$ is again upward-closed and a superset of $\mathcal{U}$. In fact, it holds that

$$\begin{aligned} \mathcal{U} \cup \mathsf{UPre}(S) &= (\lfloor \mathcal{U} \rfloor \sqcup \lfloor \mathsf{UPre}(S) \rfloor)\uparrow \\ &= (\lfloor \mathcal{U} \rfloor \sqcup \lfloor \mathsf{UPre} \rfloor(\lfloor S \rfloor)\uparrow \qquad \text{from Lemma 11.9.} \end{aligned}$$

It is easy to show by induction that $\mu X.(\mathcal{U} \cup \mathsf{UPre}(X)) = \big(\mu X.(\lfloor \mathcal{U} \rfloor \sqcup \lfloor \mathsf{UPre} \rfloor(\lfloor X \rfloor))\big)\uparrow$. Thus, $\{q_I'\} \not\sqsupseteq \mu X.(\lfloor \mathcal{U} \rfloor \sqcup \lfloor \mathsf{UPre} \rfloor(\lfloor S \rfloor))$ if and only if $q_I \notin \mu X.(\mathcal{U} \cup \mathsf{UPre}(S))$. From Proposition 11.1 and Lemmas 11.5 and 11.6 we know this is the case if and only if Eve has a winning strategy in the safety game in $\mathcal{G}'$ if and only if she wins the $\mathsf{DirFix}(\ell_{\max})$ objective in $\mathcal{G}$. □

11.3 Fix games

Since Fix games are a prefix-independent version of DirFix games, it seems logical to consider an analogue of the perfect information game from the previous section with a prefix-independent condition. Indeed, the reader might be tempted to extend the approach used to solve DirFix games by replacing the safety objective with a *co-Büchi* objective in order to solve UFix or Fix games. However, we observe that although Eve winning in the resulting game is sufficient for her to win the original Fix game, it is not necessary. Indeed, an abstract play visits states from $\mathcal{U}$ infinitely often if and only if for infinitely many i there is a concretization of the play prefix up to i which violates $\mathsf{GW}(i, \ell_{\max})$. Nevertheless, this does not imply there exists one (infinite) concretization of the play which violates $\mathsf{GW}(i, \ell_{\max})$ for infinitely many i. Figure 11.7 illustrates this phenomenon.

For the reasons stated above, we propose to solve Fix games in a different way. We first introduce the notion of *observer*. Let $\mathcal{A}$ be a deterministic parity automaton.[1] We say $\mathcal{A}$ is an *observer for the objective* V if the language of $\mathcal{A}$ is V, i.e. $\mathcal{L}_{\mathcal{A}} = V$. In [CD10], the authors show that the synchronized product of $\mathcal{G}$ and an observer for V is a parity game of perfect information which is won by Eve if and only if she wins $\mathcal{G}$. Thus, it suffices to find an algorithm to construct an observer for $\mathsf{Fix}(\ell_{\max})$ to be able to solve Fix games.

For convenience, we start by describing a non-deterministic machine that accepts as its language the complement of $\mathsf{Fix}(\ell_{\max})$. Note that all elements of $\mathsf{Fix}(\ell_{\max})$ start with the observation $\{q_I\}$ so it suffices to describe the machine that accepts any word $w \in (\Sigma \cdot \mathsf{Obs})^\omega$ such that $\{q_I\} \cdot w \in \mathsf{obs}(\mathsf{Plays}(\mathcal{G})) \setminus \mathsf{Fix}(\ell_{\max})$. The construction is similar to the one used in [CD10] to make objectives of imperfect information games visible. Intuitively, at each step of the game and after Adam has revealed the next observation we will guess his actual choice of state using non-determinism. Additionally, we shall guess whether or not a violating window starts at the next step. The state space of the automaton will therefore consist of a single state from Q, a negative integer to record the weight of the tracked window, and the length of the current open window.

Formally, let $\mathcal{N}$ be the automaton consisting of the state space $F = Q \times \{1, \ldots, \ell_{\max}\} \times \{-w_{\max} \cdot \ell_{\max}, \ldots, -1\} \cup \{\bot\}$; initial state $(q_I, 1, \bot)$; input alphabet $\Sigma' = \Sigma \times \mathsf{Obs}$; and $\Delta'' \subseteq F \times \Sigma' \times F$. The transition relation Δ'' has a transition $((p, i, n), (\sigma, o), (q, j, m))$ if $(p, \sigma, q) \in \Delta$, $q \in o$,

$$m = \begin{cases} w(p,\sigma,q) & \text{if } w(p,\sigma,q) < 0 \\ n + w(p,\sigma,q) & \text{if } n \neq \bot \wedge n + w(p,\sigma,q) < 0 \wedge i < \ell_{\max} \\ \bot & \text{otherwise,} \end{cases}$$

$$j = \begin{cases} i+1 & \text{if } m = n + w(p,\sigma,q) \\ 1 & \text{otherwise.} \end{cases}$$

We say a state $(q, i, n) \in F$ is accepting if $i = \ell_{\max}$ and $n \neq \bot$. The automaton accepts a word x if and only it has a run $(q_0, i_0, n_0)\ (\sigma_0, o_1)\ (q_1, i_1, n_1)\ (\sigma_1, o_2) \ldots$ on x such that for infinitely many j we have that (q_j, i_j, n_j) is accepting.

Proposition 11.2. *The non-deterministic Büchi automaton $\mathcal{N}$ accepts a word $\psi \in \mathsf{obs}(\mathsf{Plays}(\mathcal{G}))$ if and only if $\psi \notin \mathsf{Fix}(\ell_{\max})$.* ■

Proof.

(⇒). Assume $\mathcal{N}$ accepts ψ. Let $r = (q_0, i_0, n_0)(\sigma_0, o_1)(q_1, i_1, n_1)(\sigma_1, o_2) \ldots$ be one of the accepting runs of the automaton on ψ. By construction of $\mathcal{N}$ we have that $q_0\sigma_0 q_1 \sigma_1 \cdots \in \mathsf{obs}^{-1}(\psi)$. Let π_r denote this concrete play and $J = \{j_0, j_1, j_2, \ldots\}$ be an infinite set of indices such that $j_k < j_{k+1}$ and $(q_{j_k}, i_{j_k}, n_{j_k})$ is accepting for all $k \geq 0$. Such a sequence is guaranteed to exist since r is accepting. One can easily verify by induction on the definition of Δ'' that for all $k \geq 0$ it holds that $\pi_r \notin \mathsf{GW}(i_{j_k} - \ell_{\max}, \ell_{\max})$. It follows that $\forall m \geq 0, \exists n \geq m : \pi_r \notin \mathsf{GW}(n, \ell_{\max})$, which concludes our argument.

(⇐). Assume that $\psi = o_0\sigma_0 o_1 \sigma_1 \cdots \notin \mathsf{Fix}(\ell_{\max})$. Let $\pi = q_0\sigma_0 q_1 \sigma_1 \in \mathsf{obs}^{-1}(\psi)$ be the concrete play such that for infinitely many i it is the case

[1] We refer the reader who is not familiar with parity automata or games to [Tho95].

that $\pi \notin \mathsf{GW}(i, \ell_{\max})$. We describe the infinite run of $\mathcal{N}$ on ψ that accepts. Let $J = \{j_0, j_1, j_2, \dots\}$ be an infinite set of indices such that $j_k + \ell_{\max} < j_{k+1}$ and $\pi \notin \mathsf{GW}(j_k, \ell_{\max})$ for $k \geq 0$. The sequence is guaranteed to exist because of our choice of π. Observe that this implies there is a run $r = (q_0, i_0, n_0)$ (σ_0, o_1) (q_1, i_1, n_1) $(\sigma_1, o_2) \dots$ of the automaton where for all $k \geq 0$ we have that $n_{j_k+1} = w(q_{j_k}, \sigma_{j_k}, q_{j_k+1})$ and for all $1 < l < \ell_{\max}$ then

$$n_{j_k+l} = n_{j_k+l-1} + w(q_{j_k+l}, \sigma_{j_k+l}, q_{j_k+l+1}).$$

Furthermore, in this run it holds that for all $k \geq 0$ we have $i_{j_k+\ell_{\max}} = \ell_{\max}$. Hence, said run is such that for all $k \geq 0$ the state $(q_{j_k+\ell_{\max}}, i_{j_k+\ell_{\max}}, n_{j_k+\ell_{\max}})$ is accepting. We conclude that the automaton accepts ψ. □

At this point we determinize $\mathcal{N}$ and complement it to get a deterministic automaton with state space of size exponential in the size of $\mathcal{N}$ and parity index polynomial w.r.t. the size of Q (see [Saf88, Saf92, Pit07]). The synchronized product of $\mathcal{G}$ and the observer yields a parity game with the same size bounds. The desired result follows from the parity games' algorithm and results of [Jur00].

Theorem 11.5. *Given a WA $\mathcal{G}$, determining if Eve has a winning strategy for the $\mathsf{Fix}(\ell_{\max})$ objective can be decided in time exponential in $w_{\max}$ and the size of $\mathcal{G}$.* ■

Corollary 14. *Given a WA $\mathcal{G}$ with unary encoded weights, deciding if Eve has a winning strategy for the $\mathsf{Fix}(\ell_{\max})$ objective is* EXPTIME-*complete.* ■

11.4 UFix games

In order to determine the winner of UFix games, we proceed as in the previous section by finding a non-deterministic Büchi automaton that recognizes the set of bad abstract plays. However, in this case the situation is more complicated because a bad abstract play might arise from a violation in the *uniformity*, rather than because of a concrete path with infinitely many window violations. Figure 11.2 illustrates this issue. To overcome this, we first provide an alternative characterization of the bad abstract plays for Eve. Consider some $\psi \in \mathsf{obs}(\mathsf{Plays}(\mathcal{G}))$. We say $\pi \in \mathsf{obs}^{-1}(\psi)$ *merges with infinitely many violating paths* if for all $i \geq 0$, there are $j \geq i$, $k \geq j + \ell_{\max}$ and some $\chi \in \mathsf{obs}^{-1}(\psi[..k])$ such that $\pi[k] = \chi[k]$ and $\chi \notin \mathsf{GW}(j, \ell_{\max})$. We refer to j as the position of the violation and to k as the position of the merge. Our next result formally states the relationship between concrete plays merging for multiple violations and UFix games.

Lemma 11.10. *Given a WA $\mathcal{G}$ and $\psi \in \mathsf{obs}(\mathsf{Plays}(\mathcal{G}))$, there is $\pi \in \mathsf{obs}^{-1}(\psi)$ merging with infinitely many violating paths if and only if $\psi \notin \mathsf{UFix}(\ell_{\max})$.* ■

Proof.

($\Rightarrow$). Assume there is a $\pi \in \mathsf{obs}^{-1}(\psi)$ merging with infinitely many violating paths. We have that there are two infinite sequences of indices $J = \{j_0, j_1, \dots\}$ and $K = \{k_0, k_1, \dots\}$ such that $j_l < j_{l+1}$ and $j_l + \ell_{\max} \leq k_l$, for all $l \geq 0$, and for which we know that there is concrete path $\chi_l \in \mathsf{obs}^{-1}(\psi[..k_l])$ such that $\chi[j_l..j_l + \ell_{\max}]$ realizes an open window of length $\ell_{\max}$ and $\chi_l[k_l] = \pi[k_l]$, for all $l \geq 0$. Observe that for all $l \geq 0$ we have that $\chi_l \cdot \pi[k_l..] \in \mathsf{obs}^{-1}(\psi)$ and that $\chi_l \cdot \pi[k_l..] \notin \mathsf{GW}(j_l, \ell_{\max})$. In other words, $\forall l \geq 0, \exists \alpha \in \mathsf{obs}^{-1}(\psi), \exists m \geq l : \alpha \notin \mathsf{GW}(m, \ell_{\max})$ which implies that $\psi \notin \mathsf{UFix}(\ell_{\max})$.

($\Leftarrow$). Assume $\psi \notin \mathsf{UFix}(\ell_{\max})$. We have that there is a infinite sequence of indices $J = \{j_0, j_1, \dots\}$ such that $j_k < j_{k+1}$, for all $k \geq 0$, and for which we know there is a concrete play $\pi_k \in \mathsf{obs}^{-1}(\psi)$ such that $\pi_k \notin \mathsf{GW}(j_k - \ell_{\max}, \ell_{\max})$, for all $k \geq 0$. Observe that for all $i \geq 0$ the set $\mathsf{obs}^{-1}(\psi[i])$ is finite and bounded by $|Q|$. Thus, by Pigeonhole Principle we have that, for all $n \geq 0$ there is $\eta_n \in \{\pi_m \mid 1 \leq m \leq |Q| \cdot n\} \subseteq \mathsf{obs}^{-1}(\psi)$ which merges with at least n violating paths. Consider an arbitrary η_1. If η_1 merges with infinitely many violating paths then we are done and the claim holds. Otherwise it only merges with a finite number of violating paths, say a_1. From the previous argument we know there is an $\eta_{a'_1} \in \mathsf{obs}^{-1}(\psi)$ that merges with at least $a'_1 = a_1 + 1$. Clearly η_1 and $\eta_{a'_1}$ are disjoint at every point after a'_1, lest η_1 would merge with a new violating path. We inductively repeat the process, if $\eta_{a'_i}$ merges with infinitely many violating paths then we are done. Otherwise it only merges with some finite number of violating paths, say a_i. In that case we turn our attention to $\eta_{a'_{i+1}}$. Note that since Q is finite this process can only be done a finite number of times. Indeed, after having discarded at most $|Q| - 1$ concrete plays (which are disjoint after some finite point) it must be the case the last remaining possible concrete play has the desired property or we would have a contradiction with our assumptions. Thus, there is some concrete play $\pi \in \mathsf{obs}^{-1}(\psi)$ that merges with infinitely many violating paths. $\square$

We now construct the non-deterministic Büchi automaton $\mathcal{N}'$ that recognizes plays which contain a concrete path merging with infinitely many violating paths. The idea is that we non-deterministically keep track of two paths: one that will eventually witness a violation and then merge with the other, which ultimately serves as the witness for the path that merges with infinitely many violating paths. When the two paths merge, the automaton non-deterministically selects a new path to witness the violation. This is achieved by guessing a state in the belief set of Eve, as these states represent the end states of any concrete play consistent with the abstract play so far. To avoid the double exponential associated with taking the Reif construction before determinizing the automaton, we instead compute the belief set on-the-fly using a Mealy machine that feeds into our non-deterministic automaton. By transferring the exponential state increase to an exponential increase in the alphabet size, the overall size of the determinized automaton (after composition with the Mealy machine) will be at most singly exponential in the size of our game and W.

More specifically, denote by $\mathcal{B}$ the machine that, given $\psi = o_0\sigma_0 o_1\sigma_1 \dots \in \mathsf{obs}(\mathsf{Plays}(\mathcal{G}))$ as its input yields the infinite sequence $o_0\sigma_0 s_0 o_1\sigma_1 s_1 \dots \in (\mathsf{Obs} \cdot \Sigma \cdot \mathcal{P}(Q))^\omega$ such that $s_0 = \{q_I\}$ and for all $i \geq 0$ we have $s_{i+1} = \mathsf{post}_{\sigma_i}(s_i)$. One can easily give a definition of $\mathcal{B}$—which closely resembles a subset construction—with a state space at most exponential w.r.t. $\mathcal{G}$. Observe that $\mathcal{B}$ realizes a continuous function, in the sense that every prefix of length i of the input uniquely defines the next s_{i+1} annotation. Thus, the annotation can be done on-the-fly.

Formally, $\mathcal{N}'$ consists of the state space $F' = Q \times Q \times \{1, \dots, \ell_{\max}\} \times \{-w_{\max} \cdot \ell_{\max}, \dots, -1\} \cup \{\bot, \top\}$; initial state $(q_I, q_I, 1, \bot)$; input alphabet $\Sigma'' = \Sigma \times \mathsf{Obs} \times \mathcal{P}(Q)$; and $\Delta''' \subseteq F \times \Sigma'' \times F$. The transition relation Δ''' has a transition $((p, p', i, n), (\sigma, o, s), (q, q', j, m))$ if $(p', \sigma, q') \in \Delta$, $q \in s$, $q' \in o$,

$$m = \begin{cases} w(p,\sigma,q) & \text{if } (p,\sigma,q) \in \Delta \wedge w(p,\sigma,q) < 0 \\ n + w(p,\sigma,q) & \text{if } (p,\sigma,q) \in \Delta \wedge n \neq \bot \wedge n + w(p,\sigma,q) < 0 \wedge i < \ell_{\max} \\ \top & \text{if } (p,\sigma,q) \in \Delta \wedge (n \neq \top \vee p \neq p') \wedge n \neq \bot \wedge i = \ell_{\max} \\ \bot & \text{otherwise,} \end{cases}$$

$$j = \begin{cases} \ell_{\max} & \text{if } m = \top \\ i+1 & \text{if } m = n + w(p,\sigma,q) \\ 1 & \text{otherwise.} \end{cases}$$

We say a state $(q, q', i, n) \in F'$ is accepting if $q = q'$, $n = \top$. $\mathcal{N}'$ accepts a word x if and only it has a run (q_0, q_0', i_0, n_0) (σ_0, o_1, s_1) (q_1, q_1', i_1, n_1) $(\sigma_1, o_2, s_2) \dots$ on x such that for infinitely many j we have that (q_j, q_j', i_j, n_j) is accepting.

Proposition 11.3. *The non-deterministic Büchi automaton $\mathcal{N}'$ accepts a word $\alpha = \mathcal{B}(\psi)$, where $\psi \in \mathsf{obs}(\mathsf{Plays}(\mathcal{G}))$, if and only if $\psi \notin \mathit{UFix}(\ell_{\max})$.* ■

Proof.

($\Rightarrow$). Assume $\mathcal{N}'$ accepts α. Let $r = (q_0, q_0' i_0, n_0)$ (σ_0, o_1, s_1) $(q_1, q_1' i_1, n_1)$ $(\sigma_1, o_2, s_2) \dots$ be one of the accepting runs of the automaton on α. By construction of $\mathcal{N}'$ we have that $q_0' \sigma_0 q_1' \sigma_1 \dots \in \mathsf{obs}^{-1}(\psi)$. Let π_r denote this concrete play, $J = \{j_0, j_1, \dots\}$ and $K = \{k_0, k_1, \dots\}$ be two infinite sets of indices such that $j_l < j_{l+1}$ and $j_l + \ell_{\max} \leq k_l$, for all $l \geq 0$, and for which we know that

- $(q_{k_l}, q_{k_l}', i_{k_l}, n_{k_l})$ is accepting for all $l \geq 0$, and
- $n_{j_l + \ell_{\max}} < 0 \wedge i_{j_l + \ell_{\max}} = \ell_{\max}$.

Such sequences are guaranteed to exist since r is accepting. Assuming the correctness of $\mathcal{B}$, one can easily verify by induction on the definition of Δ''' that for all $l \geq 0$ we have that π_r, at k_l merges with a path having a violation at j_l. It follows that π_r merges with infinitely many violating paths. From Lemma 11.10 we get that $\psi \notin \mathsf{UFix}(\ell_{\max})$.

($\Leftarrow$). Assume that $\psi = o_0 \sigma_0 o_1 \sigma_1 \dots \notin \mathsf{UFix}(\ell_{\max})$. Let $\pi = q_0 \sigma_0 q_1 \sigma_1 \in \mathsf{obs}^{-1}(\psi)$ be the concrete play that merges with infinitely many violating paths (see Lemma 11.10. We describe the infinite run of $\mathcal{N}$ on $\alpha = \mathcal{B}(\psi)$ that accepts. Let $J = \{j_0, j_1, \dots\}$ and $K = \{k_0, k_1, \dots\}$ be two infinite sets of indices such that $j_l < j_{l+1}$ and $j_l + \ell_{\max} + 1 < k_l$, for all $l \geq 0$, and for which we know that there is some $\chi_l \in \mathsf{obs}^{-1}(\psi[..k_l])$ such that $\pi[k_l] = \chi_l[k_l]$ and for all $j_k < m \leq j_k + \ell_{\max} + 1$ we have $w(\chi[j_k..m]) < 0$. The sequences are guaranteed to exist because of our choice of π. Observe that this implies there is a run $r = (q_0, q_0' i_0, n_0)(\sigma_0, o_1, s_1)(q_1, q_1' i_1, n_1)(\sigma_1, o_2, s_2) \dots$ of the automaton where for all $l \geq 0$ we have that $n_{j_l+1} = w(q_{j_l}, \sigma_{j_l}, q_{j_l+1})$ and for all $1 < b < \ell_{\max}$ then

$$n_{j_l+b} = n_{j_l+b-1} + w(q_{j_l+b}, \sigma_{j_l+b}, q_{j_l+b+1}).$$

Furthermore, we have that $n_{j_l+b} = \top$ for all $\ell_{\max} \leq b \leq k_l$ and $q_{k_l} = q'_{k_l}$. Hence, said run is such that for all $l \geq 0$ the state $(q_{k_l}, q'_{k_l}, i_{k_l}, n_{k_l})$ is accepting. We conclude that the automaton accepts ψ. □

We recall that determinizing $\mathcal{N}'$ and complementing it yields an exponentially bigger deterministic automaton. Its composition with $\mathcal{B}$, itself exponentially bigger, accepts the desired set of plays and is still singly exponential in the size of the original arena and $w_{\max}$. Once more, the desired result follows from the algorithm presented in [Jur00].

Theorem 11.6. *Given a WA $\mathcal{G}$, determining if Eve has a winning strategy for the $\mathsf{UFix}(\ell_{\max})$ objective can be decided in time exponential in $w_{\max}$ and the size of $\mathcal{G}$.* ■

Corollary 15. *Given a WA $\mathcal{G}$ with unary encoded weights, deciding if Eve has a winning strategy for the $\mathsf{UFix}(\ell_{\max})$ objective is* EXPTIME*-complete.* ■

Chapter 12

Conclusion and Future Work

In this dissertation, we have summarized our work on the complexity of minimizing regret in quantitative games and computing winning observation-based strategies in quantitative games with partial observation.

We have focused on energy and mean-payoff games in our study of partial observability. However, recall that for Inf, Sup, LimInf, and LimSup one can always solve them by reduction to a Boolean game. Hence, solutions for them are implied by results found in the literature for parity games with partial observation. On the other hand, we have not considered discounted-sum games with partial observation. We comment on this in the proposed future research directions (Section 12.3) below.

12.1 Summary

For the first part of this document we have focused on regret. There, we have given algorithms to compute optimal regret-minimizing strategies for Eve when we assume she plays against an unrestricted adversary, a positional adversary, or an eloquent adversary. We have considered this problem in quantitative games with classical payoff functions Inf, Sup, LimInf, LimSup, mean payoff, and discounted sum. We have grouped all but discounted sum into prefix-independent functions. For the latter class of functions algorithms and lower bounds are almost uniform. There is one case, in particular, for which this does not hold: minimizing regret in mean-payoff games against eloquent adversaries. The latter, turns out to be undecidable in general. However, decidability can be recovered by fixing the amount of memory Eve is allowed to use. The reason why the solution for other payoff functions does not extend to mean payoff is that mean-payoff automata are not always determinizable. That is, for a given mean-payoff automaton, there might not be a deterministic mean-payoff automaton realizing the same function.

In the second part of this work we have considered energy and mean-payoff games with partial observation. These had been studied before by Degorre et al. [DDG^{+}10], but several open questions remained. In that setting, we have established tight complexity bounds for the fixed initial credit problem for energy games. We have also completed the picture regarding decidability vs. undecidability of mean-payoff games with partial observation. In particular, we have given a reduction from the halting problem for Minsky machines to determining the winner in a non-strict $\liminf$ mean-payoff game, showing that all partial-observation mean-payoff games are undecidable. Motivated by this negative result, we have defined several decidable sub-classes and studied the complexity of determining if a game is in one of those classes. We have concluded our study of quantitative games with partial observation by showing that window mean-payoff objectives are conservative approximations for mean-payoff (this was already known in the full-observation setting). We have also classified those objectives according to their decidability and given optimal algorithms to determine their winner—for the decidable cases.

12.2 Conclusion

To conclude, we have made several contributions to the game-theoretical foundations of reactive synthesis. Our work on regret has opened several new research directions which, we believe, should be investigated. We have given the first algorithms to compute regret-minimizing strategies for quantitative games played on finite arenas. The concept of regret is widely used in artificial intelligence and yet strategies with regret guarantees are mostly “template-based”, in the following sense. An algorithm is proposed and a regret bound is shown for all instances of a problem that fulfill some set of assumptions. Our approach to regret minimization has been more general and has yielded some very interesting solutions via two-player zero-sum quantitative games. Regarding our work on partial-observation games, we have closed several open problems which remained from the initial study of them. We have shown the fixed initial credit problem to be Ackermann complete. Also, we have given several decidable subclasses of mean-payoff games and studied approximations of them. As is almost always the case, though, there are several interesting open problems and new research directions, which we believe should now be investigated.

12.3 Future Work

We present here a non-exhaustive list of research problems which have arised during the preparation of this dissertation and which we believe to be interesting and well-motivated.

Concerning regret minimization, there is the question of whether one can build on our algorithms to compute a strategy which survives n steps of iterated regret minimization. This would, of course, require to consider games in which each player has its own objective.

We have also not implemented our algorithms. It would be most interesting to test them empirically and compare them against ‘regret-minimal’ strategies proposed by the artificial intelligence and machine learning communities.

In terms of computational complexity, our algorithms for discounted-sum games seem to leave space for improvement. Indeed, when the discount factor is given as part of the input (in binary) the bounds given in this work are not tight, and in some instances we only solve particular cases and not the general regret threshold problem. We believe, however, that much work on discounted-sum games and automata must take place before the behavior of the infinite-horizon discounted-sum payoff function is better understood and more properties of it are known.

Finally, there are some classes of adversary which we did not consider but which have been pointed out by other colleagues as interesting scenarios. The first of these is that of counting adversaries. Counting strategies are realizable by a 1-state Mealy machine with an unbounded counter (used to keep track of the number of turns elapsed). The second one is that of 'average adversaries'. In other words, we could assume the environment assigns a uniform probabilistic distribution to all his possible choices at every vertex. This would allow us to give 'average regret guarantees'. (The latter is open to interpretation as there might be several formalizations for the idea.)

Concerning partial-observation games, there are four main questions which we did not address in this work. First, are blind mean-payoff games undecidable if we do not restrict Eve to play finite-memory strategies only? Second, what happens in partial-observation mean-payoff games if we allow Eve to play mixed strategies? Third, are partial-observation mean-payoff games provably observably determined in ZFC? And fourth, what is the complexity of partial-observation discounted-sum games? The latter is related to some open problems for discounted-sum automata and games and will probably be the one which will remain unanswered the longest. For example, universality of discounted-sum automata and deciding the winner for a multi-dimensional discounted-sum game are problems which have not yet been solved but can be reduced to games with partial observation (see [BHO15] and references therein).

Bibliography

[ABK11] Shaull Almagor, Udi Boker, and Orna Kupferman. What's decidable about weighted automata? In *ATVA*, pages 482–491. Springer, 2011.

[AG11] Krzysztof R. Apt and Erich Grädel. *Lectures in game theory for computer scientists.* Cambridge University Press, 2011.

[AHK02] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *J. ACM*, 49(5):672–713, 2002.

[AK86] Krzysztof R. Apt and Dexter Kozen. Limits for automatic verification of finite-state concurrent systems. *Information Processing Letters*, 22(6):307–309, 1986.

[AKL10] Benjamin Aminof, Orna Kupferman, and Robby Lampert. Reasoning about online algorithms with weighted automata. *ACM Transactions on Algorithms*, 2010.

[AOS+16] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *CoRR*, abs/1606.06565, 2016.

[AR14] Benjamin Aminof and Sasha Rubin. First cycle games. In *Strategic Reasoning*, volume 146 of *EPTCS*, pages 83–90, 2014.

[BBF+12] Aaron Bohy, Véronique Bruyère, Emmanuel Filiot, Naiyong Jin, and Jean-François Raskin. Acacia+, a tool for LTL synthesis. In *CAV*, pages 652–657. Springer, 2012.

[BCD+08] Dietmar Berwanger, Krishnendu Chatterjee, Laurent Doyen, Thomas A. Henzinger, and Sangram Raje. Strategy construction for parity games with imperfect information. In *CONCUR*, pages 325–339, 2008.

[BCD+11] Lubos Brim, Jakub Chaloupka, Laurent Doyen, Raffaella Gentilini, and Jean-François Raskin. Faster algorithms for mean-payoff games. *Formal Methods in System Design*, 38(2):97–118, 2011.

[BCH+16] Romain Brenguier, Lorenzo Clemente, Paul Hunter, Guillermo A. Pérez, Mickael Randour, Jean-François Raskin, Ocan Sankur, and Mathieu Sassolas. Non-zero sum games for reactive synthesis. In *LATA*, volume 9618 of *LNCS*, pages 3–23. Springer, 2016.

[BCW$^+$09] Dietmar Berwanger, Krishnendu Chatterjee, Martin De Wulf, Laurent Doyen, and Thomas A. Henzinger. Alpaga: A tool for solving parity games with imperfect information. In *TACAS*, volume 5505 of *LNCS*. Springer, 2009.

[BD08] Dietmar Berwanger and Laurent Doyen. On the power of imperfect information. In *FSTTCS*, pages 73–82, 2008.

[Bel82] David E. Bell. Regret in decision making under uncertainty. *Operations Research*, 30(5):961–981, 1982.

[Ber07] Dietmar Berwanger. Admissibility in infinite games. In *STACS*, LNCS 4393, pages 188–199. Springer, 2007.

[BFK08] Adam Brandenburger, Amanda Friedenberg, and H. Jerome Keisler. Admissibility in games. *Econometrica*, 76(2), 2008.

[BFL$^+$08] Patricia Bouyer, Ulrich Fahrenberg, Kim Guldstrand Larsen, Nicolas Markey, and Jirí Srba. Infinite runs in weighted timed automata with energy constraints. In Franck Cassez and Claude Jard, editors, *FORMATS*, volume 5215 of *LNCS*, pages 33–47. Springer, 2008.

[BFRR14] Véronique Bruyère, Emmanuel Filiot, Mickael Randour, and Jean-François Raskin. Meet your expectations with guarantees: Beyond worst-case synthesis in quantitative games. In *STACS*, LIPIcs 25, pages 199–213. Schloss Dagstuhl - LZI, 2014.

[BH14] Udi Boker and Thomas A. Henzinger. Exact and approximate determinization of discounted-sum automata. *LMCS*, 10(1), 2014.

[BHO15] Udi Boker, Thomas A. Henzinger, and Jan Otop. The target discounted-sum problem. In *LICS*, pages 750–761. IEEE Computer Society, 2015.

[BK08] Christel Baier and Joost-Pieter Katoen. *Principles of model checking.* MIT Press, 2008.

[BMR14] Véronique Bruyère, Noémie Meunier, and Jean-François Raskin. Secure equilibria in weighted games. In *CSL-LICS*, pages 26:1–26:26, 2014.

[BRS14] Romain Brenguier, Jean-François Raskin, and Mathieu Sassolas. The complexity of admissibility in omega-regular games. In *CSL-LICS*, pages 23:1–23:10. ACM, 2014.

[BRS15] Romain Brenguier, Jean-François Raskin, and Ocan Sankur. Assume-admissible synthesis. In *CONCUR*, volume 42 of *LIPIcs*, pages 100–113. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[BSV04] Henrik Björklund, Sven Sandberg, and Sergei Vorobyov. Memoryless determinacy of parity and mean payoff games: a simple proof. *TCS*, 310(1):365–378, 2004.

[CD10] Krishnendu Chatterjee and Laurent Doyen. The complexity of partial-observation parity games. In *LPAR*, pages 1–14. Springer, 2010.

[CD12] Krishnendu Chatterjee and Laurent Doyen. Partial-observation stochastic games: How to win when belief fails. In *LICS*, pages 175–184. IEEE Computer Society, 2012.

[CdAHS03] Arindam Chakrabarti, Luca de Alfaro, Thomas A. Henzinger, and Mariëlle Stoelinga. Resource interfaces. In Rajeev Alur and Insup Lee, editors, *EMSOFT*, volume 2855 of *LNCS*, pages 117–133. Springer, 2003.

[CDH10] Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Quantitative languages. *ACM Transactions on Computational Logic*, 11(4), 2010.

[CDHR06] Krishnendu Chatterjee, Laurent Doyen, Thomas A Henzinger, and Jean-François Raskin. Algorithms for omega-regular games with imperfect information. In *CSL*, pages 287–302, 2006.

[CDRR13] Krishnendu Chatterjee, Laurent Doyen, Mickaël Randour, and Jean-François Raskin. Looking at mean-payoff and total-payoff through windows. In *ATVA*, pages 118–132. Springer, 2013.

[Cha07] Krishnendu Chatterjee. Concurrent games with tail objectives. *TCS*, 388(1):181–198, 2007.

[CKS81] Ashok K. Chandra, Dexter Kozen, and Larry J. Stockmeyer. Alternation. *J. ACM*, 28(1):114–133, 1981.

[CR15] Lorenzo Clemente and Jean-François Raskin. Multidimensional beyond worst-case and almost-sure problems for mean-payoff objectives. In *LICS*, pages 257–268. IEEE, 2015.

[DDG+10] Aldric Degorre, Laurent Doyen, Raffaella Gentilini, Jean-François Raskin, and Szymon Toruńczyk. Energy and mean-payoff games with imperfect information. In *CSL*, pages 260–274, 2010.

[DF11] Werner Damm and Bernd Finkbeiner. Does it pay to extend the perimeter of a world model? In *FM*, volume 6664 of *LNCS*, pages 12–26. Springer, 2011.

[dGRdB+15] Stijn de Gouw, Jurriaan Rot, Frank S. de Boer, Richard Bubel, and Reiner Hähnle. Openjdk's java.utils.collection.sort() is broken: The good, the bad and the worst case. In *CAV (1)*, volume 9206 of *LNCS*, pages 273–289. Springer, 2015.

[DJW97] Stefan Dziembowski, Marcin Jurdziński, and Igor Walukiewicz. How much memory is needed to win infinite games? In *LICS*, pages 99–110. IEEE Computer Society, 1997.

[DR09] Laurent Doyen and Jean-François Raskin. Antichains for the automata-based approach to model-checking. *LMCS*, 5(1), 2009.

[DR10] Laurent Doyen and Jean-François Raskin. Antichain algorithms for finite automata. In *TACAS*, volume 6015 of *LNCS*, pages 2–22. Springer, 2010.

[EJ91] E. Allen Emerson and Charanjit S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In *STOCS*, pages 368–377. IEEE, 1991.

[EJ99] E. Allen Emerson and Charanjit S. Jutla. The complexity of tree automata and logics of programs. *SIAM J. Comput.*, 29(1):132–158, 1999.

[EM79] Andrzej Ehrenfeucht and Jan Mycielski. Positional strategies for mean payoff games. *International Journal of Game Theory*, 8:109–113, 1979.

[ET98] Tali Eilam-Tzoreff. The disjoint shortest paths problem. *Discrete Applied Mathematics*, 85(2):113–138, 1998.

[Fae09] Marco Faella. Admissible strategies in infinite games over graphs. In *MFCS*, volume 5734 of *LNCS*, pages 307–318. Springer, 2009.

[FFSS11] Diego Figueira, Santiago Figueira, Sylvain Schmitz, and Philippe Schnoebelen. Ackermannian and primitive-recursive bounds with dickson's lemma. In *LICS*, pages 269–278. IEEE Computer Society, 2011.

[FGR10] Emmanuel Filiot, Tristan Le Gall, and Jean-François Raskin. Iterated regret minimization in game graphs. In *MFCS*, volume 6281 of *LNCS*, pages 342–354. Springer, 2010.

[FW98] Matt Fairtlough and Stanley S Wainer. Hierarchies of provably recursive functions. *Handbook of proof theory*, 137:149–207, 1998.

[Gel14] Marcus Gelderie. *Strategy machines: representation and complexity of strategies in infinite games.* PhD thesis, RWTH Aachen University, 2014.

[Gen14] Raffaella Gentilini. A note on the approximation of mean-payoff games. *Information Processing Letters*, 114(7):382–386, 2014.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, 1979.

[GO10] Hugo Gimbert and Youssouf Oualhadj. Probabilistic automata on finite words: Decidable and undecidable problems. In *ICALP*, volume 6199 of *LNCS*, pages 527–538. Springer, 2010.

[Grä04] Erich Grädel. Positional determinacy of infinite games. In *STACS*, pages 4–18. Springer, 2004.

[HM15] Axel Haddad and Benjamin Monmege. Why Value Iteration Runs in Pseudo-Polynomial Time for Discounted-Payoff Games. Technical note, Université libre de Bruxelles, 2015.

[HP06] Thomas A. Henzinger and Nir Piterman. Solving games without determinization. In *CSL*, pages 395–410. Springer, 2006.

[HP12] Joseph Y. Halpern and Rafael Pass. Iterated regret minimization: A new solution concept. *Games and Economic Behavior*, 74(1):184–207, 2012.

[HPR14] Paul Hunter, Guillermo A. Pérez, and Jean-François Raskin. Mean-payoff games with partial-observation - (extended abstract). In *Reachability Problems*, pages 163–175, 2014.

[HPR16a] Paul Hunter, Guillermo A. Pérez, and Jean-François Raskin. Minimizing regret in discounted-sum games. In *CSL*, LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. To appear.

[HPR16b] Paul Hunter, Guillermo A. Pérez, and Jean-François Raskin. Reactive synthesis without regret. *Acta Informatica*, pages 1–37, 2016.

[JBB+14] Swen Jacobs, Roderick Bloem, Romain Brenguier, Rüdiger Ehlers, Timotheus Hell, Robert Könighofer, Guillermo A. Pérez, Jean-François Raskin, Leonid Ryzhyk, Ocan Sankur, Martina Seidl, Leander Tentrup, and Adam Walker. The first reactive synthesis competition (SYNTCOMP 2014). Technical Report 1506.08726, arXiv, 2014.

[JLS15] Marcin Jurdziński, Ranko Lazic, and Sylvain Schmitz. Fixed-dimensional energy games are in pseudo-polynomial time. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *ICALP*, volume 9135 of *LNCS*, pages 260–272. Springer, 2015.

[JSL08] Marcin Jurdzinski, Jeremy Sproston, and François Laroussinie. Model checking probabilistic timed automata with one or two clocks. *LMCS*, 4(3), 2008.

[Jur98] Marcin Jurdziński. Deciding the winner in parity games is in $\mathrm{UP}\cap\mathrm{coUP}$. *Information Processing Letters*, 68(3):119–124, 1998.

[Jur00] Marcin Jurdziński. Small progress measures for solving parity games. In *STACS*, volume 1770 of *LNCS*, pages 290–301. Springer, 2000.

[Kec95] Alexander S. Kechris. *Classical descriptive set theory*, volume 156. Springer-Verlag New York, 1995.

[Kle56] S. C. Kleene. Representation of events in nerve nets and finite automata. *Automata Studies*, 1956.

[Kri03] Thomas Krieger. On pareto equilibria in vector-valued extensive form games. *Mathematical Methods of Operations Research*, 58(3):449–458, 2003.

[KV00] Orna Kupferman and Moshe Y Vardi. Synthesis with incomplete informatio. *Advances in Temporal Logic*, 16:109–127, 2000.

[Kwi07] Marta Z. Kwiatkowska. Quantitative verification: models, techniques and tools. In Ivica Crnkovic and Antonia Bertolino, editors, *SIGSOFT*, pages 449–458. ACM, 2007.

[LW70] Martin H. Löb and Stanley S. Wainer. Hierarchies of number-theoretic functions. i. *Archive for Mathematical Logic*, 13(1):39–51, 1970.

[Mar75] Donald A. Martin. Borel determinacy. *The annals of Mathematics*, 102(2):363–371, 1975.

[Mar98] Donald A. Martin. The determinacy of blackwell games. *The Journal of Symbolic Logic*, 63(4):1565–1581, 1998.

[Min67] Marvin Lee Minsky. *Computation: Finite and Infinite Machines.* Prentice-Hall, 1967.

[MS88] Donald A. Martin and John R. Steel. Projective determinacy. *National Academy of Sciences of the United States of America*, 85(18):6582, 1988.

[N^+50] John F. Nash et al. Equilibrium points in n-person games. *Nat. Acad. Sci. USA*, 36(1):48–49, 1950.

[OR94] Martin J. Osborne and Ariel Rubinstein. *A course in game theory.* MIT press, 1994.

[Pap03] Christos H. Papadimitriou. *Computational complexity.* John Wiley and Sons Ltd., 2003.

[Pit07] Nir Piterman. From nondeterministic Büchi and Streett automata to deterministic parity automata. *LMCS*, 3(3), 2007.

[PP04] Dominique Perrin and Jean-Eric Pin. *Infinite words: automata, semigroups, logic and games*, volume 141. Academic Press, 2004.

[PP06] Nir Piterman and Amir Pnueli. Faster solutions of Rabin and Streett games. In *LICS*, pages 275–284, 2006.

[PY86] Christos H. Papadimitriou and Mihalis Yannakakis. A note on succinct representations of graphs. *Information and Control*, 71(3):181 – 185, 1986.

[PY91] Christos H. Papadimitriou and Mihalis Yannakakis. Shortest paths without a map. *TCS*, 84(1):127–150, 1991.

[Rab69] Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969.

[RCDH07] Jean-François Raskin, Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Algorithms for omega-regular games with imperfect information. *LMCS*, 3(3), 2007.

[Rei84] John H. Reif. The complexity of two-player games of incomplete information. *Journal of computer and system sciences*, 29(2):274–301, 1984.

[Rou09] Stéphane Le Roux. Acyclic preferences and existence of sequential nash equilibria: A formal and constructive equivalence. In *TPHOLs*, volume 5674 of *LNCS*, pages 293–309. Springer, 2009.

[RP15] Stéphane Le Roux and Arno Pauly. Weihrauch degrees of finding equilibria in sequential games. In *CiE*, volume 9136 of *LNCS*, pages 246–257. Springer, 2015.

[Saf88] Shmuel Safra. On the complexity of ω-automata. In *FOCS*, pages 319–327. IEEE, 1988.

[Saf92] Shmuel Safra. Exponential determinization for omega-automata with strong-fairness acceptance condition (extended abstract). In *STOC*, pages 275–282, 1992.

[Sch16] Sylvain Schmitz. Complexity hierarchies beyond elementary. *ACM Transactions on Computation Theory*, 8(1):3, 2016.

[SHM$^+$16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[SM73] Larry J. Stockmeyer and Albert R. Meyer. Word problems requiring exponential time. In *STOC*, pages 1–9. ACM, 1973.

[SS12] Sylvain Schmitz and Philippe Schnoebelen. Algorithmic aspects of wqo theory. Lecture Notes, 2012.

[Tho95] Wolfgang Thomas. On the synthesis of strategies in infinite games. In *STACS*, pages 1–13. Springer, 1995.

[Ves06] René Vestergaard. A constructive approach to sequential nash equilibria. *Information Processing Letters*, 97(2):46–51, 2006.

[vNM07] John von Neumann and Oskar Morgenstern. *Theory of games and economic behavior*. Princeton university press, 2007.

[Wai70] Stanley S Wainer. A classification of the ordinal recursive functions. *Archive for Mathematical Logic*, 13(3):136–153, 1970.

[WET15] Min Wen, Rüdiger Ehlers, and Ufuk Topcu. Correct-by-synthesis reinforcement learning with temporal logic constraints. In *IROS*, pages 4983–4990. IEEE, 2015.

[ZJBP08] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In *NIPS*, pages 905–912, 2008.

[ZP96] Uri Zwick and Mike Paterson. The complexity of mean payoff games on graphs. *TCS*, 158(1):343–359, 1996.

Index