Time-optimal Winning Strategies in Infinite Games

(Zeit-optimale Gewinnstrategien in unendlichen Spielen)

von

Martin Zimmermann

Matrikelnummer 250702

DIPLOMARBEIT

im Studiengang Informatik

vorgelegt der Fakultät für
Mathematik, Informatik und Naturwissenschaften der
Rheinisch-Westfälischen Technischen Hochschule Aachen

angefertigt am
LEHRSTUHL FÜR INFORMATIK VII
Logik und Theorie diskreter Systeme
Prof. Dr. Dr.h.c. Wolfgang Thomas

Ich versichere hiermit, dass ich diese Arbeit selbständig verfasst habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht.

Aachen, den 22. Januar 2009 (Martin Zimmermann)

Abstract

Infinite Games are an important tool in the synthesis of finite-state controllers for reactive systems. The interaction between the environment and the system is modeled by a finite graph. The specification that has to be satisfied by the controlled system is translated into a winning condition on the infinite paths of the graph. Then, a winning strategy is a controller that is correct with respect to the given specification. Winning strategies are often finitely described by automata with output.

While classical optimization of synthesized controllers focuses on the size of the automaton we consider a different quality measure. Many winning conditions allow a natural definition of waiting times that reflect periods of waiting in the original system. We investigate time-optimal strategies for Request-Response Games, Poset Games - a novel type of infinite games that extends Request-Response Games - and games with winning conditions in Parametric Linear Temporal Logic. Here, the temporal operators of classical Temporal Logics can be subscribed with free variables that represent bounds on the satisfaction. Then, one is interested in winning strategies with respect to optimal valuations of the free variables. The optimization objective, maximization respectively minimization of the variable values, depends on the formula.

For Request-Response Games and Poset Games, we prove the existence of time-optimal finite-state winning strategies. For games with winning conditions in Parametric Linear Temporal Logic, we prove that optimal winning strategies are computable for solitary games.

Contents

To Heike

Chapter 1

Introduction

Game Theory first came to prominence in 1944 with the seminal book [43] by Morgenstern and von Neumann, and was subsequently developed into an interdisciplinary field, which covers economics, biology, political science, and computer science amongst other fields. Nowadays, there is an abundance of different games tailored to model certain aspects of nature, society, or mathematics that can in general be categorized along the following dimensions.

Players: From single-player games over games with finitely many players to games with infinitely many players.

Moves: The players choose their actions simultaneously or sequentially.

Duration: From a single action to games of infinite duration.

Payoffs: Are the gains and losses of the players balanced or not.

Cooperation: Do the players aim to maximize their own payoff or the payoff of a coalition they belong to.

Information: Do the players observe all actions or is there uncertainty about the state of the game.

Classically, there are two ways to represent games. A game in *normal form* consists of a finite set of actions and a payoff function for each player. Every player chooses an action (without knowledge of the choices of the other players) and the payoff is determined from the tuple of chosen actions. The hand game Rock Paper Scissors can be seen as a game in normal form. Two players simultaneously pick one of the following actions: rock, paper, or scissors. The outcome is determined by the combination of the choices: rock blunts scissors, paper covers rock, and scissors cut paper. *Strategies* can either be *pure*, i.e., each player picks an action, or *mixed*, i.e., each player picks a probability distribution over the set of actions. Combinations of strategies such that it is disadvantageous for every player to unilaterally change her strategy, so called *Equilibria*, are a key concept of game theory. An early milestone of Game Theory is the existence of equilibria in mixed strategies due to Nash [42].

A game in *extensive form* is played on a tree of finite height, where a token is placed at the root. Each non-terminal node belongs to one of the players who decides to which successor the token is moved if it reaches that node. The terminal nodes are marked with a payoff for each player. Tic Tac Toe, for example, can easily be modeled in extensive form.

Two other flavors of games can be found in logics. *Game semantics* define satisfaction of a formula φ in a structure $\mathfrak{A}$ as a game between two players, Verifier, who tries to prove $\mathfrak{A} \models \varphi$, and Falsifier, who tries to prove $\mathfrak{A} \not\models \varphi$. The positions of the game are the subformulae of φ. For example, a disjunction $\varphi_1 \vee \varphi_2$ is satisfied, if one of the φ_i is satisfied. Therefore, disjunctions belong to Verifier and she can move to either one of the disjuncts. A universally quantified formula $\forall x\varphi$ is unsatisfiable if there is an element a of $\mathfrak{A}$ such that $\varphi[x\backslash a]$ is unsatisfiable. Consequently, the positions $\forall x\varphi$ belong to Falsifier and he can move to $\varphi[x\backslash a]$ for every a. For First-Order logic, these *Model-Checking Games* are Reachability Games with finite plays only, while Model-Checking Games for Fixed-Point logics have infinite plays. Parity Games, for example, are the Model-Checking Games for the modal μ-Calculus [18, 17]. This explains the importance of Parity Games for verification and the ongoing efforts in determining the exact complexity of solving them. We return to this question later on.

Model Comparison Games on the other hand are a tool to prove inexpressibility results. Such a game is played on two structures $\mathfrak{A}$ and $\mathfrak{B}$. The existence of a winning strategy for the first player is equivalent to the indistinguishability of $\mathfrak{A}$ and $\mathfrak{B}$ for the logic under consideration. Thus, if one can exhibit a winning strategy for two structures that differ in some property, then this property cannot be expressed in the corresponding logic. Ehrenfeucht-Fräissé Games [14, 21] are used to prove that two structures satisfy exactly the same First-Order formulae, while the modal μ-Calculus (and all the Temporal Logics it subsumes) is bisimulation invariant, a notion which can be defined by a game as well.

The type of game we are interested in arose in the 1960s from the theory of automata on infinite objects. The breakthrough of this theory was Büchi's proof that the Monadic Second Order Theory of the natural numbers with the successor relation is decidable [5]. This was the first in a long line of decidability results which culminated in Rabin's Tree Theorem [47], which proves the decidability of the Monadic Second Order Logic of the Binary Tree. All these results rely on the expressive equivalence of a logic and an appropriate automaton model, for example Büchi Automata on infinite words and Rabin Tree Automata on infinite trees for the results mentioned above. This equivalence is also the origin of Model-Checking [10], where a specification is translated into an automaton. Then, the well-developed techniques of automata theory can be applied to check whether the system satisfies the specification. This approach is implemented in numerous verification tools and is well-established in industrial applications.

Seeing automata from a different angle, Church posed in 1957 the following problem inspired by the synthesis of switching circuits [8]: given a specification on two bitstreams, an input stream and an output stream, compute a finite automaton with output that computes for every input stream an output stream, such that the pair of streams satisfies the specification. This problem can easily be turned into a game between an environment and a controller. The players choose bits in alternation, the first player constructs the input sequence, and the second player constructs the output sequence while trying to satisfy the specification. The controller's winning strategy, if it is finitely describable, can be converted into a circuit that is guaranteed to satisfy the specification on its behavior. Büchi and Landweber [6] were able to solve the *synthesis problem* for specifications in Monadic Second Order Logic. A key element of their proof was the determinization of automata on infinite words due to McNaughton [39].

Subsequently, the setting was generalized to model systems for program verification. The main characteristics of this type of games are reactiveness, the program competes against an adversarial environment, and infinite duration, which models the non-terminating nature of controllers, drivers and operating systems. An *infinite game* is played on a graph whose set of vertices is partitioned into the positions of Player 0 and 1. The two players construct an infinite path by moving a token through the graph. After ω moves, the winner is determined. A strategy for Player i is a lookup table that contains a successor for each finite play ending in a vertex of Player i. In this setting, Martin was able to prove his far-reaching Borel Determinacy Theorem [37]: in every game whose set of winning paths for Player 0 is a Borel set, one of the players has a winning strategy. The Borel sets are induced by a topology on infinite words and encompass almost all winning conditions discussed in the literature. These conditions are often acceptance conditions of automata on infinite objects that were transfered to games, for example the Büchi and Co-Büchi, Muller, Rabin, Streett, and Parity conditions (confer [25] for details). Others, like Reachability, Safety, and Request-Response conditions [54] are defined for games, but are less compelling as acceptance conditions for automata.

Arguably, the most important question in the theory of infinite games concerns the complexity of solving Parity Games. Positional determinacy [17, 41] places the problem in $\mathbf{NP} \cap \mathbf{coNP}$ and Jurdziński [31] improved this to $\mathbf{UP} \cap \mathbf{coUP}$. Several algorithms have been presented ([32, 4, 48] amongst others) over the course of time, the best with subexponential running time [33]. Another promising algorithm was presented by Vöge and Jurdziński [52], whose time complexity is still an open problem. It is unknown whether there exists a polynomial-time algorithm.

Muller, Rabin, Streett, and Request-Response Games cannot be won with positional strategies, but with finite-state strategies. The quality of such a strategy is typically measured in the size of its memory. Matching upper and lower bounds hold for Muller, Rabin, and Streett Games [13] and for Request-Response Games [54]. Since the lower bounds are worst case results, one can still try to minimize the size of a given winning strategy. As finite-state strategies are nothing more than automata with output, or transducers, one can apply automata minimization techniques to the underlying memory structure. However, a minimization algorithm only minimizes the size of the memory structure, but does not change the strategy. While this implies the correctness of the minimization, it cannot always yield a smallest strategy as it might be necessary to change the strategy. Winning strategies for Request-Response Games and Staiger-Wagner Games (weak Muller Games) can be minimized by altering the strategy the automaton implements [27].

In this work, we are interested in another kind of quality. Many winning conditions allow an intuitive definition of waiting times.

Reachability Games: The number of moves before the token reaches one of the designated vertices.

Büchi Games: The number of moves between the visits of designated vertices.

Co-Büchi Games: The number of moves before the token reaches the designated vertices for good.

Parity Games: The number of moves between the visits of a vertex of maximal color that is seen infinitely often.

Request-Response Games: The number of moves between a request and the subsequent response.

In some games there are infinitely many waiting times that have to be aggregated. It is natural to ask whether there are optimal strategies that minimize the (aggregated) waiting times. For Reachability, Büchi, and Co-Büchi Games, the attractor-based algorithms [25] compute optimal winning strategies. For Parity Games, there are two optimizations goals: firstly to maximize the highest even color that is seen infinitely often (without visiting a higher odd color infinitely often), and secondly to minimize the intervals between visits of that color. The strategy improvement algorithm from [52] computes optimal strategies in the following sense: its first priority is to maximize the highest even color that is seen infinitely often and its second priority is to minimize the waiting times between the visits of that color.

With the same goal in mind, finitary Parity and Streett Games are introduced [7]. In these games, Player 0 wins only if there is a bound on the waiting times. Determinacy and memory requirements can be proven and algorithms solving finitary games (see also [28]) exist.

Time-optimal strategies for Request-Response Games were first investigated by Wallmeier [53] and extended by Horn et. al. [29]. If the waiting times are aggregated by taking the average mean of the accumulated waiting times, then optimal finite-state strategies exist and can be computed effectively.

For games with winning conditions in Linear Temporal Logic (LTL), one can introduce bounded operators to obtain a quantitative notion of satisfaction. The bounded eventuality $\mathbf{F}_{\leq 10}\varphi$ should be read as "φ holds within the next 10 steps". There is a long history of extensions of LTL with such constructs. We want to mention two such logics.

Parametric Linear Temporal Logic [1] adds an abundance of additional operators to LTL, parameterized both by constants and variables. Satisfaction is then defined with respect to a variable valuation and turns into an optimization problem: find the best valuation such that φ holds with respect to that valuation. The Satisfiability problem and Model-Checking can be decided as well as several natural optimization problems.

Prompt-LTL [35] adds the operator prompt-eventually $\mathbf{F}_p$ to LTL with the following semantics. The formula $\mathbf{F}_p\varphi$ is satisfied if there is some fixed k such that φ is satisfied within k steps. Model-Checking and realizability, a game-theoretic problem in spirit of Church's synthesis problem, are decidable for Prompt-LTL.

In another line of research Gimbert and Zielonka [22, 23] determined necessary and sufficient conditions for games that have optimal positional strategies. This class includes Parity Games, Mean-Payoff Games [15], and Discounted Payoff Games [55]. Also, there are tight connections between discounted games and discounted versions of the μ-Calculus [24, 11, 20].

Outline

Following this introduction, we fix our notation and present the components of infinite games in Chapter 2. Also, we introduce some useful tools for solving games and state basic results about infinite games which we will rely on throughout this thesis. The rest of this work is devoted to the definition and computation of waiting time based quality measures for strategies in infinite games.

We begin with Request-Response Games, for which a framework for defining time-optimal strategies was already developed by Wallmeier [53]. Waiting times are defined in the natural way and the quality of a play is measured in the long term average of the waiting times. This framework is extended slightly by Horn et. al. [29] and time-optimal finite-state winning strategies can be computed in both frameworks. We present another proof in Chapter 3 for two reasons. Firstly, the presentation in [29] is erroneous. We fix and complement this proof. By doing this, we obtain a flexible proof technique that can be applied to other winning conditions as well.

We do this for Poset Games in Chapter 4, a novel winning condition extending the Request-Response winning condition: responses are replaced by a poset of events and a request is responded by an embedding of these events. This allows to express more complicated conditions, for example problems from planning and scheduling that cannot be modeled by a Request-Response winning condition. After covering some basic notions of Order Theory, Poset Games are defined formally and solved by a reduction to Büchi Games. To complete this introductory treatment, we prove that this reduction is asymptotically optimal. Then, we turn our attention to the quality of a strategy. Waiting times are given by the length of the interval between the request and the completion of the corresponding embedding. The main theorem of this chapter states the existence of time-optimal finite-state winning strategies. We close the chapter by discussing the differences between the frameworks when applied to Request-Response Games.

For Request-Response Games and Poset Games, the existence of time-optimal finite-state winning strategies is proved in Chapter 3 and Chapter 4, respectively. Both proofs consist of two steps. The first one is to show that for every strategy of small value there is another strategy, whose value is equal or smaller that bounds the waiting times. To this end, a strategy improvement operator is defined that deletes costly loops. This operator is applied infinitely often and the limit of these strategies bounds all waiting times. The value of an optimal strategy can be bounded from above by the value of the finite-state strategy obtained from the reduction to Büchi Games, which implies that an optimal strategy bounds the waiting times. In the second step, the game is reduced to a Mean-Payoff Game, such that the values of the two games coincide. Since optimal strategies for Mean-Payoff Games can be computed, this suffices to prove that time-optimal finite-state winning strategies exist and can be computed effectively.

In Chapter 5 winning conditions in Parametric Linear Temporal Logic are analyzed. Following prior work on satisfiability and Model-Checking of Parametric Linear Temporal logic by Alur et. al. [1], we focus on two fragments, one obtained by adding the operator $\mathbf{F}_{\leq x}$, the other by adding $\mathbf{G}_{\leq y}$ to LTL, where x and y are free variables. Then, one can ask whether Player 0 wins a game with respect to some, infinitely many, or all variable valuations. We adapt the techniques developed for Satisfiability and Model-Checking to infinite games and are able to prove that these questions and several natural optimization problems can be solved effectively for solitary games. Some of the decision problems are also decidable for two-player games, but most decision problems and all optimization problems remain open for two-player games.

Chapter 6 concludes this work and gives some hints to open problems and future research. The memory of the optimal strategies computed in Chapter 3 and 4 is very large, but we give some pointers that should allow a dramatic reduction of the size. Another interesting aspect is the trade-off between the size of a finite-state winning strategy and its quality. Concerning games with winning conditions in Parametric Linear Temporal Logic, the major open question is the analysis of two-player games. We hint at some problems one encounters when trying to adapt the techniques for solitary games to two-player games.

Acknowledgements

I would like to thank Prof. Wolfgang Thomas for his guidance and encouragement throughout my work. Further, I am grateful for an inspiring discussion with Florian Horn about Request-Response Games, which ultimately lead to the proof presented in Chapter 3. I would also like to thank Prof. Wolfgang Thomas and Prof. Joost-Pieter Katoen for examining this thesis.

Furthermore, I am very grateful to Ingo Felscher, Michael Holtmann, Bernd Puchala, Torsten Sattler, Alex Spelten, and Nadine Wacker for proofreading this thesis, especially to Nadine and Torsten, who mastered the whole work. Also, I am thankful to Volker Kamin, Martin Plücker, and Torsten Sattler for their companionship during the years studying together. You guys made things a lot easier.

I am deeply indebted to my parents whose support made my academic studies possible. Thank you for everything.

Finally, I want to thank Nadine Wacker for challenging me every day. And for just being the way you are. Mavericks 95, Bucks 93.

Chapter 2

Preliminaries

In this section, we fix our notation and state some results. After beginning with the most basic definitions in Section 2.1, we introduce automata on infinite words in Section 2.2 and Linear Temporal Logic in Section 2.3. Afterwards, we focus on games by defining the different components of a game: arenas and plays in Section 2.4, winning conditions in Section 2.5, and strategies in Section 2.6. Finite-state strategies and game reductions, two closely related concepts, are presented in Section 2.7. While game reductions expand the arena, a strategy restricts the set of possible plays in an arena. The various kinds of restrictions are introduced in Section 2.8. To conclude the chapter, we state some results about games in Section 2.9, on which we rely throughout this thesis. This chapter only covers concepts important to our cause, defining and computing time-optimal winning strategies for infinite games. Hence, many interesting aspects of infinite games are not discussed here. For a more thorough introduction to the theory of automata on infinite words and infinite games, we refer the reader to [25, 51, 50].

2.1 Numbers, Words, and Trees

The set of non-negative integers is denoted by $\mathbb{N} = \{0, 1, 2, \ldots\}$. For a natural number n let $[n] = \{1, \ldots, n\}$, especially $[0] = \emptyset$. For a set S denote the powerset of S by 2^S and the cardinality of S by $|S|$. An *enumeration* of a finite set S is a bijection $e : [|S|] \to S$.

An *alphabet* Σ is a finite, non-empty set of *letters* or *symbols*, Σ^* is the set of (finite) *words* over Σ, $\varepsilon \in \Sigma^*$ denotes the *empty word*, and $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$ is the set of *non-empty words* over Σ. Concatenation of words is denoted by juxtaposition, and the length of a word w is denoted by $|w|$. An *infinite word* α over Σ is denoted by $\alpha = \alpha_0\alpha_1\alpha_2\ldots$, where $\alpha_n \in \Sigma$ for all $n \in \mathbb{N}$. The set of all infinite words over Σ is Σ^ω. A *language* L is a set $L \subseteq \Sigma^*$ or $L \subseteq \Sigma^\omega$.

A word x is a *prefix* of y, written $x \sqsubseteq y$, if $y = xz$ for some word z, and x is a *proper prefix* of y, written $x \sqsubset y$, if $x \sqsubseteq y$ and $x \neq y$. Similarly, a word x is a prefix of an ω-word α, written $x \sqsubset \alpha$, if $\alpha = x\beta$ for some ω-word β. Given $L \subseteq \Sigma^*$ or $L \subseteq \Sigma^\omega$ let $\mathrm{Pref}(L)$ be the set of prefixes of the (ω-) words in L. A word x is an *infix* of a finite word y if there exist words y_1, y_2 such that $y = y_1 x y_2$, and x is an infix of an infinite word α, if x is infix of some prefix of α. A word x is a *subword* of y if $x = x_1 \cdots x_n$ and there exist words $y_0, \ldots, y_n \in \Sigma^*$ such that $y = y_0 x_1 y_1 \cdots y_{n-1} x_n y_n$. An ω-word α is *ultimately periodic*, if $\alpha = xy^\omega$ for some $x, y \in \Sigma^*$.

For an ω-word $\alpha = \alpha_0 \alpha_1 \alpha_2 \ldots \in \Sigma^\omega$, let

$$\mathrm{Occ}(\alpha) = \{a \in \Sigma \mid \exists n : \alpha_n = a\}$$

be the *occurrence set* of α and

$$\mathrm{Inf}(\alpha) = \{a \in \Sigma \mid \exists^\omega n : \alpha_n = a\}$$

be the *infinity set* of α. The occurrence set of a finite word w is defined analogously.

Given a word $w' = wx$, the *left quotient of w from w'* is $w^{-1}w' = x$. This operation can be lifted to languages $L \subseteq \Sigma^*$ and $w \in \Sigma^*$. The *left quotient of w from L* is $w^{-1}L = \{x \in \Sigma^* \, | wx \in L\}$.

A prefix-closed set of words $L \subseteq \Sigma^*$ induces the *tree* $\mathfrak{T}(L) = (L, E)$ where the set of edges is given by $E = \{(w, wa) \mid w, wa \in L, a \in \Sigma\}$. Similarly, $K \subseteq \Sigma^\omega$ induces a tree with vertex set $V = \mathrm{Pref}(K)$. However, there might exist infinite paths of $\mathfrak{T}(\mathrm{Pref}(K))$ that are not in K. Given a tree $\mathfrak{T}(L)$ for some $L \subseteq \Sigma^*$ and $w \in L$, let $\mathfrak{T}(L)\upharpoonright_w = \mathfrak{T}(w^{-1}L)$ be the *subtree of $\mathfrak{T}(L)$ rooted in w*.

Given a sequence $(w_n)_{n\in\mathbb{N}}$ of finite words such that $w_n \sqsubset w_{n+1}$ for all n, $\lim_{n\to\infty} w_n$ denotes the unique ω-word α, such that $w_n \sqsubset \alpha$ for every n.

Let $(f_n)_{n\in\mathbb{N}}$ be a sequence of functions $f_n : A \to B$ and $f : A \to B$. We say that $(f_n)_{n\in\mathbb{N}}$ *converges* to the *limit* f, $\lim_{n\to\infty} f_n = f$, if

$$\forall a \in A \, \exists n_a \in \mathbb{N} \, \forall n \geq n_a : f_n(a) = f(a).$$

Otherwise, $(f_n)_{n\in\mathbb{N}}$ *diverges*. Obviously, if $(f_n)_{n\in\mathbb{N}}$ converges, then the limit f is uniquely determined.

2.2 Automata on Infinite Words

A *Büchi automaton* $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ consists of a finite set Q of *states*, a finite alphabet Σ, an *initial state* $q_0 \in Q$, a *transition relation* $\Delta \subseteq Q \times \Sigma \times Q$, and a set $F \subseteq Q$ of *final states*. A *Muller Automaton* is a tuple $\mathcal{M} = (Q, \Sigma, q_0, \Delta, \mathcal{F})$ where Q, Σ, q_0, and Δ are as above and $\mathcal{F} \subseteq 2^Q$.

A *run* of an automaton on an ω-word $\alpha = \alpha_0\alpha_1\alpha_2 \ldots \in \Sigma^\omega$ is an infinite sequence $\rho = \rho_0\rho_1\rho_2 \ldots \in Q^\omega$ such that $\rho_0 = q_0$ and $(\rho_n, \alpha_n, \rho_{n+1}) \in \Delta$ for all n. The automaton $\mathcal{A}$ *accepts* α, if there exists a run ρ of $\mathcal{A}$ on α such that $\mathrm{Inf}(\rho) \cap F \neq \emptyset$, and $\mathcal{M}$ *accepts* α, if there exists a run ρ of $\mathcal{M}$ on α such that $\mathrm{Inf}(\rho) \in \mathcal{F}$. The *language* of an automaton, $L(\mathcal{A})$ and $L(\mathcal{M})$, respectively, is the set of ω-words accepted by the corresponding automaton.

An automaton is *deterministic*, if for every $(s, a) \in Q \times \Sigma$ there is exactly one s' such that $(s, a, s') \in \Delta$, i.e., Δ is equivalent to a function $\delta : Q \times \Sigma \to Q$. Non-deterministic Büchi Automata and deterministic Muller Automata (and many other types of automata) accept the same class of languages, the so-called *regular languages*.

2.3 Linear Temporal Logic

Let P be a set of atomic propositions. A *labeled graph* $G = (V, E, l)$ consists of a set V of *vertices*, a set $E \subseteq V \times V$ of edges, and a *labeling function* $l : V \to 2^P$.

The set LTL of *Linear Temporal Logic* formulae is defined inductively by

- $p \in \text{LTL}$ and $\neg p \in \text{LTL}$ if $p \in P$,
- $\varphi \wedge \psi \in \text{LTL}$ and $\varphi \vee \psi \in \text{LTL}$ if $\varphi, \psi \in \text{LTL}$, and
- $\mathbf{X}\varphi \in \text{LTL}$, $\varphi\mathbf{U}\psi \in \text{LTL}$, and $\varphi\mathbf{R}\psi \in \text{LTL}$ if $\varphi, \psi \in \text{LTL}$.

Additionally, we define $\mathbf{tt} = p \vee \neg p$ and $\mathbf{ff} = p \wedge \neg p$ for some $p \in P$, $\mathbf{F}\varphi = \mathbf{ttU}\varphi$ and $\mathbf{G}\varphi = \mathbf{ffR}\varphi$. The *size of* φ, denoted by $|\varphi|$, is defined as the number of distinct subformulae of φ.

Let $G = (V, E, l)$ be a labeled graph and $\rho = \rho_0\rho_1\rho_2 \ldots \in V^\omega$ a path in G. The *satisfaction relation* $\models$ is defined inductively by

- $(\rho, n) \models p$ iff $p \in l(\rho_n)$,
- $(\rho, n) \models \neg p$ iff $p \notin l(\rho_n)$,
- $(\rho, n) \models \varphi \wedge \psi$ iff $(\rho, n) \models \varphi$ and $(\rho, n) \models \psi$,
- $(\rho, n) \models \varphi \vee \psi$ iff $(\rho, n) \models \varphi$ or $(\rho, n) \models \psi$,
- $(\rho, n) \models \mathbf{X}\varphi$ iff $(\rho, n + 1) \models \varphi$,
- $(\rho, n) \models \varphi\mathbf{U}\psi$ iff $\exists k \geq 0$ such that $(\rho, n + k) \models \psi$ and $\forall l < k : (\rho, n + l) \models \varphi$, and
- $(\rho, n) \models \varphi\mathbf{R}\psi$ iff $\forall k \geq 0$: either $(\rho, n + k) \models \psi$ or $\exists l < k$ such that $(\rho, n + l) \models \varphi$.

Finally, define $\rho \models \varphi$, if $(\rho, 0) \models \varphi$. In this case, we say ρ is a *model* of φ. Although, we only allow negation of atomic propositions LTL is closed under negation, due to the duality of $\wedge$ and $\vee$, by $\neg\mathbf{X}\varphi \equiv \mathbf{X}\neg\varphi$, and $\neg(\varphi\mathbf{U}\psi) \equiv (\neg\varphi)\mathbf{R}(\neg\psi)$. A formula $\varphi \in \text{LTL}$ defines the (regular [5]) language $L(\varphi)$ of ω-words over the alphabet 2^P, consisting of the ω-words that are a model of φ.

2.4 Arenas, Plays, and Games

The games we consider in this thesis are turn-based two-player games of perfect information and infinite duration. They are played on a directed graph equipped with a partition of the vertices that determines the positions of Player 0 and Player 1. The positions of Player 0 are drawn as circles whereas Player 1's positions are drawn as rectangles. For pronominal convenience, we assume that Player 0 is female while Player 1 is male.

To begin a play a token is placed at an initial vertex. Then, at every step the player, at whose position the token sits, moves the token along an edge to another vertex. This way, the players build up a play, an infinite sequence of vertices. After ω steps the outcome of the play is determined. Most of the games we consider are zero-sum games, i.e., one of the players wins a play while the other one loses it. In the following, we introduce the basic ingredients of infinite games.

An *arena* $G = (V, V_0, V_1, E)$ consists of a finite, directed graph (V, E) where the vertex set V is the disjoint union of the *positions of Player* 0, V_0, and the *positions of Player* 1, V_1. The *moves* are given by the *edge relation* $E \subseteq V \times V$, which we require to contain at least one outgoing edge for every vertex. A *solitary arena* for Player i is an arena (V, V_0, V_1, E) such that $V_{1-i} = \emptyset$. Equivalently, one could allow every vertex in V_{1-i} to have exactly one successor. Then, Player $1-i$ has only one legal move at every position, which could also be made by Player i.

We disallow dead ends in order to avoid the nuisance of defining the winner of finite plays. This does not impose a restriction since every arena with dead ends can be equipped with a sink. However, the modification has to respect the intended winning condition. This depends on the actual winning condition and the way finite plays ending in dead ends are scored. Finally, we consider only finite arenas. Some results we present do only hold for these, because they rely on counting arguments. However, all definitions are applicable to infinite arenas without modifications.

A *play* $\rho = \rho_0\rho_1\rho_2\dots$ is an infinite sequence of vertices such that $(\rho_n, \rho_{n+1}) \in E$ for all n. In proofs, we have to deal with finite prefixes of plays. All suitable definitions for infinite plays are defined for finite prefixes accordingly, but are not explicitly stated.

Example 2.1. To illustrate the definitions above, consider the arena $G = (V, V_0, V_1, E)$ depicted in Figure 2.1, which is our running example throughout this chapter. The positions of Player 0 are $V_0 = \{s_2, s_4\}$, Player 1's positions are $V_1 = \{s_0, s_1, s_3, s_5\}$. The arcs denote the possible moves. A possible play in G is $\rho = s_0s_2s_4s_5s_3s_1^\omega$.

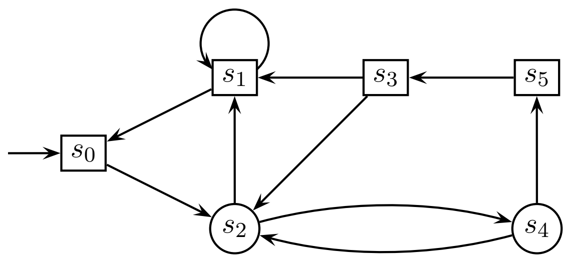

Figure 2.1: The arena G

A *game* $\mathcal{G} = (G, \varphi)$ consists of an arena $G = (V, V_0, V_1, E)$ and a *winning condition* φ specifying the set of *winning plays* $\text{Win} \subseteq V^\omega$. A play ρ is won by Player 0, if $\rho \in \text{Win}$. Otherwise, it is won by Player 1. In the following, we often consider games with designated initial vertex: an *initialized game* (G, s, φ) consists of a game (G, φ) and a vertex s of G. In such a game, all plays start in s. Finally, a *solitary game* for Player i is a game played in a solitary arena for Player i.

2.5 Winning Conditions

The most general outcome of a play is a payoff for each player. This is modeled by *payoff functions* p_i for Player i specifying the payoff $p_i(\rho)$ for every play ρ. Yet, most games, both in real life and in mathematics, are antagonistic: the gain of a player is the loss of the other player. Mathematically speaking, the payoffs for every play sum up to zero. Accordingly, such games are called *zero-sum games*. For most of our purposes we can even abstract from an actual payoff and just declare a winner for each play. Then, the other player loses the play. This corresponds to the general definition of a game from above employing a set of winning plays for Player 0. We stick to this with the exception of one type of games that is introduced at the end of this section.

While it would suffice to specify the set Win of winning plays for Player 0 directly, games typically employ a winning condition φ that defines Win indirectly. The advantage lies in the intuitive nature of these winning conditions, which simplifies reasoning about those games considerably.

Nevertheless, we begin in an abstract setting: the *Borel Hierarchy* consists of ω-languages and is build up from a class of basic languages, comprised of the open sets $Z \cdot \Sigma^\omega$ for $Z \subseteq \Sigma^*$, by applying complementation and countable union. To avoid delving into topology, we refer the curious reader to [34]. We just observe that every regular language is a *Borel set*, a set contained in the Borel Hierarchy.

$\mathcal{G}$ is a Borel Game [37], if the set of winning plays is a Borel set. This broad class, which encompasses most of the zero-sum games that can be found in literature, is of interest, since it enjoys useful properties. As it is often easy to show that a set of winning plays, defined by a winning condition φ, is Borel, this is often the first step in the analysis of a new type of game.

As hinted at above, the set of winning plays is typically given implicitly by a winning condition φ, oftentimes as requirements on $\text{Occ}(\rho)$ or $\text{Inf}(\rho)$. Several conditions have been investigated in the literature. We introduce here only those that are of interest to our work.

Büchi Games: $\varphi = F \subseteq V$ and $\rho \in \text{Win}$ iff $\text{Inf}(\rho) \cap F \neq \emptyset$.

Generalized Büchi Games: $\varphi = (F_j)_{j=1,\ldots,k}$ where $F_j \subseteq V$ and $\rho \in \text{Win}$ iff $\text{Inf}(\rho) \cap F_j \neq \emptyset$ for all $j \in [k]$.

Request-Response Games: $\varphi = (Q_j, P_j)_{j=1,\ldots,k}$, where $Q_j, P_j \subseteq V$, and $\rho \in \text{Win}$ iff $\forall j \, \forall n (\rho_n \in Q_j \rightarrow \exists n' \geq n : \rho_{n'} \in P_j)$.

Muller Games: $\varphi = \mathcal{F} \subseteq 2^V$ and $\rho \in \text{Win}$ iff $\text{Inf}(\rho) \in \mathcal{F}$.

Parity Games: $\varphi = c : V \to [k]$, for some k, and $\rho \in \text{Win}$ iff $\max(\text{Inf}(c(\rho)))$ is even. Here, c is a *coloring* of the arena's vertices. For a play $\rho = \rho_0\rho_1\rho_2 \ldots \in V^\omega$ let $c(\rho) = c(\rho_0)c(\rho_1)c(\rho_2)\ldots$.

LTL Games: $\varphi \in \text{LTL}$ and $\rho \in \text{Win}$ iff $\rho \models \varphi$. Here, the graph underlying the arena is assumed to be labeled.

It is easy to show that all conditions introduced above define a Borel set of winning plays. An important subclass of Borel Games are regular games, games whose winning plays for Player 0 form a regular language.

Example 2.2. We continue Example 2.1 by specifying two games with arena G.

(i) The initialized Borel Game $\mathcal{G}_1 = (G, s_0, \text{Win})$ with $\text{Win} = \{\rho \mid \{s_1, s_3\} \subseteq \text{Occ}(\rho)\}$. The play $s_0s_2s_4s_5s_3s_1^\omega$ is won by Player 0 whereas the play $(s_0s_2s_1)^\omega$ is won by Player 1.

(ii) The Büchi Game $\mathcal{G}_2 = (G, F)$ with $F = \{s_1\}$. The play $s_3(s_2s_4)^\omega$ is won by Player 1; however, Player 0 can do better with $s_3(s_2s_1s_0)^\omega$, for example, a play that she wins.

A class of games that does not fit into the framework outlined above are Mean-Payoff Games [15]. They are not zero-sum and the payoffs are defined by an integer-labeling of the edges. The players try to maximize respectively minimize certain means of the sum of labels seen on a play. A (initialized) *Mean-Payoff Game* $\mathcal{G} = (G, s, d, l)$ consists of an arena $G = (V, V_0, V_1, E)$, an initial vertex $s \in V$, $d \in \mathbb{N}$, and a function $l : E \to \{-d, \ldots, d\}$ assigning integer labels to the edges. For a play $\rho = \rho_0\rho_1\rho_2\ldots$ define the *gain* $v_0(\rho)$ for Player 0,

$$v_0(\rho) = \liminf_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} l(\rho_i, \rho_{i+1}),$$

and the *loss* $v_1(\rho)$ for Player 1,

$$v_1(\rho) = \limsup_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} l(\rho_i, \rho_{i+1}).$$

Player 0 tries to maximize $v_0(\rho)$ whereas Player 1 tries to minimize $v_1(\rho)$. Obviously, the gain for Player 0 is never higher than the loss for Player 1.

2.6 Strategies and Positional Strategies

After introducing the way infinite games are played, we now consider the most important and interesting aspect of games: how to choose the next move? In general, this decision may depend on the history of the play, the sequence of moves made by the players so far. Strategies are introduced in this general sense, but typically a more restrictive notion suffices, which limits the amount of information about the history that is used to determine the next move. The most extreme choice is to use no information at all, i.e., the choice of the next move depends only on the current position of the token. It turns out that several games can be won with those simple strategies.

Let $G = (V, V_0, V_1, E)$ be an arena. A *strategy* for Player i is a (partial) mapping $\sigma : V^*V_i \to V$ such that $(s, \sigma(ws)) \in E$ for all $w \in V^*$ and all $s \in V_i$. The set of all strategies for Player i (in a fixed arena) is denoted by Γ_i. We denote strategies for Player 0 (and the indefinite Player i) by σ and strategies for Player 1 by τ.

A play $\rho_0\rho_1\rho_2\dots$ is *played according to* σ or is *consistent with* σ, if $\rho_{n+1} = \sigma(\rho_0 \dots \rho_n)$ for every $\rho_n \in V_i$. The strategy σ is a *winning strategy* for Player i from $s \in V$, if every play starting in s that is played according to σ is won by Player i.

The *winning region* W_i of Player i is

$$W_i = \{s \in V \mid \text{Player } i \text{ has a winning strategy from } s\}.$$

Obviously, we have $W_0 \cap W_1$ for every game. A game is *determined*, if $W_0 \cup W_1 = V$, i.e., from every vertex, one of the players has a winning strategy. An initialized game (G, s, φ) is *won* by Player i, if she has a winning strategy from s. Otherwise she loses the game. Determinacy means that exactly one of the players wins (G, s, φ) while the other one loses the game. This is trivially true for a single play, but not for a game in general. Nevertheless, all zero-sum games we consider in this thesis are determined. Note that our definition of determinacy is not applicable to games that are not zero-sum; however, the definition can be extended accordingly. *Solving* a game $\mathcal{G}$ amounts to determining W_0 and W_1 and corresponding winning strategies.

A rather restrictive notion of strategies is obtained by prohibiting the use of any information about the history of the play. The choice of the next move only depends on the vertex the token is at. Nevertheless, these strategies suffice to win many kinds of games. Formally, we say a strategy σ for Player i is *positional*, if $\sigma(ws) = \sigma(w's)$ for all $w, w' \in V^*$ and all $s \in V_i$. Hence, a positional strategy is fully specified by a mapping that assigns a successor to every vertex in V_i. We use both representations interchangeably.

Example 2.3. Again, we continue Example 2.1 by defining winning strategies for the two games defined in Example 2.2.

(i) For $\mathcal{G}_1$, with initial vertex s_0, the winning condition for Player 0 requires the token to visit both s_1 and s_3. Therefore, Player 0 cannot move the token to s_1 as soon as it reaches s_2 after the first move of Player 1. Rather, she has to move it via $s4$ to s_5 first, from which Player 1 has only one choice, namely, to move the token to s_3. From there, he can either move the token to s_1 directly, and lose thereby, or move it to s_2, from where Player 0 can move it to s_1, and again win the play in doing so. The remainder of the play is irrelevant, then. Thus, we define the strategy σ_1 for Player 0 for a finite play w and $s \in V_0$ as follows

$$\sigma_1(ws) = \begin{cases} s_4 & \text{if } s = s_2 \text{ and } s_3 \notin \mathrm{Occ}(w) \\ s_1 & \text{if } s = s_2 \text{ and } s_3 \in \mathrm{Occ}(w) \\ s_5 & \text{if } s = s_4 \end{cases}$$

As reasoned above, σ_1 is a winning strategy for Player 0 in $\mathcal{G}_1$. Note however, that σ_1 is not positional.

(ii) For the Büchi Game $\mathcal{G}_2$, the winning condition requires the token to visit s_1 infinitely often. We define a positional strategy σ_2 by $\sigma_2(s_2) = s_1$ and $\sigma_2(s_4) = s_5$. It is easy to verify that every play consistent with σ_2 visits s_1 infinitely often. Thus, σ_2 is a winning strategy from every vertex for Player 0 in $\mathcal{G}_2$.

2.7 Finite-State Strategies and Game Reductions

A compromise between a positional strategy and a strategy with infinite domain is a *strategy with memory*. Here, the decision about the next move does not take into account the complete history, but some abstraction of it. Thus, two different histories can have the same abstraction and therefore share the same next move. Oftentimes, there are only finitely many abstractions; hence, the strategy is realizable with *finite memory*. Nevertheless, we give all definitions as general as possible.

Let $G = (V, V_0, V_1, E)$ be an arena. A *memory structure* $\mathfrak{M} = (M, \mathrm{init}, \mathrm{update})$ for G consists of a non-empty set M of *memory states*, an *initialization function* $\mathrm{init} : V \to M$, and an *update function* $\mathrm{update} : M \times V \to M$.

The *memory content reached after* $w = w_0 \dots w_n \in V^+$, $\mathrm{update}^*(w)$, is defined inductively by $\mathrm{update}^*(w_0) = \mathrm{init}(w_0)$ and

$$\mathrm{update}^*(w_0 \dots w_n) = \mathrm{update}(\mathrm{update}^*(w_0 \dots w_{n-1}), w_n).$$

A function $\mathrm{next} : V_i \times M \to V$ is a *next-move function* for Player i, if $(s, \mathrm{next}(s, m)) \in E$ for all $m \in M$ and $s \in V_i$. A next-move function induces a *strategy with memory* $\mathfrak{M}$ for Player i via $\sigma(w_0 \dots w_n) = \mathrm{next}(w_n, \mathrm{update}^*(w_0 \dots w_n))$.

We call the strategy σ *finite-state*, if M is finite. We call $|M|$ the *size of* $\mathfrak{M}$ and (slightly abusive) the size of an induced (finite-state) strategy.

Remark 2.4. *If $|M| = 1$, then is σ positional.*

Example 2.5. The winning strategy σ_1 from Example 2.3 for $\mathcal{G}_1$ from Example 2.2 can be implemented as finite-state strategy. The memory is used to remember whether s_3

has been visited on the play so far, and the choice of the next move at s_2 depends on that. Formally, we define $\mathfrak{M} = (M, \text{init}, \text{update})$ as follows.

- $M = \{0, 1\}$,
- $\text{init}(s_0) = 0$ (init can be defined arbitrarily for all other states, since all plays start at the initial vertex s_0), and
- $\text{update}(b, s) = \begin{cases} 1 & \text{if } b = 0 \text{ and } s = s_3 \\ 0 & \text{otherwise} \end{cases}$.

The next-move function is given by

$$\text{next}(s, b) = \begin{cases} s_5 & \text{if } s = s_4 \\ s_4 & \text{if } s = s_2 \text{ and } b = 0 \\ s_1 & \text{if } s = s_2 \text{ and } b = 1 \end{cases}.$$

The strategy σ implemented by $\mathfrak{M}$ and the function next is the winning strategy σ_1 from Example 2.3.

An arena $G = (V, V_0, V_1, E)$ and a memory structure $\mathfrak{M}$ for G induce the arena

$$G \times \mathfrak{M} = (V \times M, V_0 \times M, V_1 \times M, E_{\text{update}})$$

where $E_{\text{update}} = \{((s, m), (s', m')) \mid (s, s') \in E \text{ and } m' = \text{update}(m, s')\}$. Every play $\rho' = (\rho_0, m_0)(\rho_1, m_1)(\rho_2, m_2)\ldots$ in the expanded arena $G \times \mathfrak{M}$ has a unique *projected play* $\rho = \rho_0\rho_1\rho_2\ldots$ in G. Conversely, every play $\rho = \rho_0\rho_1\rho_2\ldots$ in G has a unique *expanded play* $\rho' = (\rho_0, m_0)(\rho_1, m_1)(\rho_2, m_2)\ldots$ in $G \times \mathfrak{M}$ induced by $\mathfrak{M}$ via $m_0 = \text{init}(\rho_0)$ and $m_{n+1} = \text{update}(m_n, \rho_{n+1})$.

Often, a game (G, φ) can be *reduced* to a game with expanded arena $G \times \mathfrak{M}$ with winning condition φ' for a suitable memory structure $\mathfrak{M}$. A reduction allows to modify strategies for the expanded arena to strategies for the original arena. If games with winning condition φ' are easier to solve than games with φ, then a reduction is oftentimes the natural way to solve games with winning condition φ.

Let $\mathcal{G} = (G, \varphi)$ and $\mathcal{G}' = (G', \varphi')$ be games, and $\mathfrak{M}$ a memory structure for G. We say $\mathcal{G}$ is *reducible* to $\mathcal{G}'$ via $\mathfrak{M}$, written $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$, if $G' = G \times \mathfrak{M}$ and every play in $\mathcal{G}'$ is won by the same player that wins the projected play of $\mathcal{G}$.

If $|\mathfrak{M}| = 1$, then G is isomorphic to $G \times \mathfrak{M}$. We say that $\mathcal{G}$ is *equivalent* to $\mathcal{G}'$, if $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$ for some memory structure $\mathfrak{M}$ with $|\mathfrak{M}| = 1$. Informally speaking, if $\mathcal{G}$ and $\mathcal{G}'$ are equivalent, then φ' is at least as expressive as φ.

Example 2.6. Let G be an arena.

(i) Let (G, F) be a Büchi Game. The Parity Game (G, c) with $c(s) = 1$, if $s \notin F$, and $c(s) = 2$, if $s \in F$, is equivalent to (G, F).

(ii) Let (G, c) be a Parity Game. The Muller Game $(G, \mathcal{F})$ is equivalent to (G, c), if $F \in \mathcal{F}$ iff $\max\{c(s) \mid s \in F\}$ is even.

Our main interest in game reductions is their usage in solving games. But before we state the reduction theorem, we define the *composition* of memory structures. This allows us to give a more general statement than the one that is used in most reductions, which is an easy corollary. Let $\mathfrak{M} = (M, \mathrm{init}, \mathrm{update})$ be a memory structure for an arena G and let $\mathfrak{M}' = (M', \mathrm{init}', \mathrm{update}')$ be a memory structure for $G \times \mathfrak{M}$. We obtain a memory structure $\mathfrak{M}'' = \mathfrak{M} \times \mathfrak{M}' = (M'', \mathrm{init}'', \mathrm{update}'')$ for G where $M'' = M \times M'$, $\mathrm{init}''(s) = (\mathrm{init}(s), \mathrm{init}'(s, \mathrm{init}(s)))$, and

$$\mathrm{update}''((m, m'), s) = (\mathrm{update}(m, s), \mathrm{update}'(m', (s, \mathrm{update}(m, s)))).$$

Theorem 2.7 (Reduction Theorem)**.** *Let $\mathfrak{M} = (M, \mathrm{init}, \mathrm{update})$ be a memory structure for an arena G and $\mathfrak{M}' = (M', \mathrm{init}', \mathrm{update}')$ be a memory structure for $G \times \mathfrak{M}$. Furthermore, let $\mathcal{G}$ and $\mathcal{G}'$ be games with arena G respectively $G \times \mathfrak{M}$. If $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$ and Player i has a winning strategy σ' with memory $\mathfrak{M}'$ for $\mathcal{G}'$ from position $(s_0, \mathrm{init}(s_0))$, then she also has a winning strategy σ with memory $\mathfrak{M} \times \mathfrak{M}'$ for $\mathcal{G}$ from s_0.*

Proof. Let σ' be induced by $\mathrm{next}' : (V_i \times M) \times M' \to V \times M$. We need to define a next-move function $\mathrm{next} : V_i \times (M \times M') \to V$ such that it induces a winning strategy σ for $\mathcal{G}$. For $(s, m) \in V_i \times M$ and $m' \in M'$ such that $\mathrm{next}'((s, m), m') = (s', m'')$, let $\mathrm{next}(s, (m, m')) = s'$.

Let $\rho = \rho_0\rho_1\rho_2 \ldots$ be a play according to σ in G such that $\rho_0 = s_0$. Furthermore, let $\rho' = (\rho_0, m_0)(\rho_1, m_1)(\rho_2, m_2) \ldots$ and $\rho'' = (\rho_0, (m_0, m'_0))(\rho_1, (m_1, m'_1))(\rho_2, (m_2, m'_2)) \ldots$ be the unique expanded plays in $G' = G \times \mathfrak{M}$ respectively in $G \times (\mathfrak{M} \times \mathfrak{M}')$. By definition, we have $(\rho_0, m_0) = (s_0, \mathrm{init}(s_0))$. Thus, if ρ' is played according to σ', then it is won by Player i. Since $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$, ρ is then won by Player i as well. Hence, σ is a winning strategy for Player i from s for $\mathcal{G}$.

So, it remains to show that ρ' is consistent with σ': let $(\rho_n, m_n) \in V_i \times M$. We have $\rho_{n+1} = \sigma(\rho_0 \ldots \rho_n) = \mathrm{next}(\rho_n, (m_n, m'_n))$. Since $\mathrm{next}(\rho_n, (m_n, m'_n))$ is the first component of $\mathrm{next}'((\rho_n, m_n), m'_n)$, we have $\mathrm{next}'((\rho_n, m_n), m'_n) = (\rho_{n+1}, m)$ for some $m \in M$. Since $((\rho_n, m_n), (\rho_{n+1}, m))$ is an edge of $G \times \mathfrak{M}$, we have $m = \mathrm{update}(m_n, \rho_{n+1}) = m_{n+1}$. Hence, we have $\sigma'((\rho_0, m_0) \ldots (\rho_n, m_n)) = \mathrm{next}'((\rho_n, m_n), m'_n) = (\rho_{n+1}, m_{n+1})$ for all $(\rho_n, m_n) \in V_i \times M$. Therefore, ρ' is played according to σ'. □

Corollary 2.8. *Let $G' = G \times \mathfrak{M}$, σ' be a strategy in G and σ be the induced strategy in G as above. Furthermore, let ρ and ρ' be plays in G consistent with σ respectively in G' consistent with σ'.*

(i) The expanded play of ρ is consistent with σ'.

(ii) The projected play of ρ' is consistent with σ.

Another important consequence of the theorem is concerned with the case of positional winning strategies for $\mathcal{G}'$.

Corollary 2.9. *If $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$ and Player i has a positional winning strategy for $\mathcal{G}'$ from position $(s, \mathrm{init}(s))$, then she has a winning strategy with memory $\mathfrak{M}$ for $\mathcal{G}$ from s.*

The last corollary relates the winning regions of the two games.

Corollary 2.10. *If $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$ and W_i' is the winning region of Player i in $\mathcal{G}'$, then $W_i = \{s \in V \mid (s, \mathrm{init}(s)) \in W_i'\}$ is the winning region of Player i in $\mathcal{G}$.*

Example 2.11. Let $\mathcal{G} = (G, \varphi)$ be a regular game, i.e., the set of winning plays for Player 0, $\mathrm{Win} \subseteq V^\omega$, is a regular language. Then, Win is the language accepted by some deterministic Muller Automaton $\mathcal{M} = (Q, V, q_0, \delta, \mathcal{F})$ [25]. Define $\mathfrak{M} = (Q, \mathrm{init}, \mathrm{update})$ where $\mathrm{init}(s) = q_0$ for all $s \in V$ and $\mathrm{update} = \delta$. Finally, define the Muller Game $\mathcal{G}' = (G \times \mathfrak{M}, \mathcal{F}')$ where

$$\{(v_1, q_1), \ldots, (v_n, q_n)\} \in \mathcal{F}' \Leftrightarrow \{q_1, \ldots, q_n\} \in \mathcal{F}.$$

Then, we have $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$, i.e., every regular ame can be reduced to a Muller Game.

2.8 Unravelings and Restricted Arenas

Given an arena $G = (V, V_0, V_1, E)$ and an initial vertex $s \in V$ let, $\mathfrak{T}_{G,s} = (V^*, V_0^*, V_1^*, E^*)$ be the *unraveling* of G from s where V^* is the set of finite plays of G starting in s, V_i^* contains exactly those plays in V^* that end in a vertex of V_i, and $(ws', ws's'') \in E^*$ iff $ws's'' \in V^*$ and $(s', s'') \in E$. A play $\rho^* = (\rho_0)(\rho_0\rho_1)(\rho_0\rho_1\rho_2)\ldots$ in $\mathfrak{T}_{G,s}$ starting in s is uniquely determined by the sequence $\rho = \rho_0\rho_1\rho_2\ldots$, which is a play in G starting in s. Conversely, every play in G determines a unique play in $\mathfrak{T}_{\mathcal{G},s}$. Thus, we denote plays in $\mathfrak{T}_{G,s}$ by the respective play in G.

Also, every winning condition for G can be translated into a winning condition for the unraveled arena such that the winner of a play ρ in G and its counterpart ρ^* in $\mathfrak{T}_{G,s}$ are the same. Finally, a strategy σ^* for $\mathfrak{T}_{G,s}$ can be transformed into a strategy σ in G by $\sigma(\rho_0 \ldots \rho_n) = \sigma^*((\rho_0) \ldots (\rho_0 \ldots \rho_n))$. The reverse transformation, from a strategy σ in G into a strategy σ^* for $\mathfrak{T}_{G,s}$, is given by $\sigma^*((\rho_0) \ldots (\rho_0 \ldots \rho_n)) = \sigma(\rho_0 \ldots \rho_n)$.

Thus, we can reason about an arena G or its unraveling $\mathfrak{T}_{G,s}$ and translate the results back and forth. The main benefit of reasoning about the unraveling instead of the original arena is that $\mathfrak{T}_{G,s}$ is a tree, thus every strategy σ^* is positional. This considerably simplifies the discussion about arbitrary strategies for an arena.

For a strategy σ for Player i in G let $\mathfrak{T}_{G,s}^\sigma$ be the *restriction of* $\mathfrak{T}_{G,s}$ to those plays that are consistent with σ. Every vertex in V_i^* has exactly one child in the restricted unraveling. Conversely, every subtree obtained from $\mathfrak{T}_{G,s}$ by deleting all but one child (and the subtrees rooted in these vertices) of every vertex in V_i^* induces a strategy for

Player i. For strategies σ and τ for Player 0 respectively 1 let $\mathfrak{T}_{G,s}^{\sigma,\tau}$ be the restriction to the unique play that is consistent with σ and τ.

For a finite play w denote the subtree of $\mathfrak{T}_{G,s}$ rooted in w by $\mathfrak{T}_{G,s}\upharpoonright_w$. The definitions for $\mathfrak{T}_{G,s}^{\sigma}\upharpoonright_w$ and $\mathfrak{T}_{G,s}^{\sigma,\tau}\upharpoonright_w$ are analogous.

For positional strategies in G, we do not need to take the detour via the unraveling to define restricted arenas: let $G = (V, V_0, V_1, E)$ be an arena and $\sigma : V_i \to V$ a positional strategy for Player i. The *restriction of* G to σ is $G\upharpoonright_\sigma = (V, V_0, V_1, E')$ where

$$E' = \{(s, \sigma(s)) \mid s \in V_i\} \cup \{(s, s') \in E \mid s \in V_{1-i}\}.$$

Note that the unraveling of $G\upharpoonright_\sigma$ from s is $\mathfrak{T}_{G,s}^{\sigma}$. Also, $G\upharpoonright_\sigma$ is a solitary game for Player $1-i$ (in the wider sense discussed above).

For $s \in V$, and strategies σ and τ for Player 0 respectively Player 1, we define the play $\rho(s, \sigma, \tau) = \rho_0\rho_1\rho_2\ldots$ where $\rho_0 = s$ and

$$\rho_{n+1} = \begin{cases} \sigma(\rho_0 \ldots \rho_n) & \text{if } \rho_n \in V_0 \\ \tau(\rho_0 \ldots \rho_n) & \text{if } \rho_n \in V_1 \end{cases}.$$

Again, $\rho(s, \sigma, \tau)$ is equal to the only play in $\mathfrak{T}_{G,s}^{\sigma,\tau}$. If both strategies are positional, then $\rho(s, \sigma, \tau)$ is the only play in $(G\upharpoonright_\sigma)\upharpoonright_\tau$ starting in s. If G is a solitary arena for Player 0, then we have $\rho(s, \sigma, \tau) = \rho(s, \sigma, \tau')$ for all strategies τ and τ' for Player 1, i.e., the strategy of Player 1 is irrelevant. Therefore, we can write $\rho(s, \sigma)$ for short. We use the same notation for Player 1, if there is no ambiguity.

Remark 2.12. *Let σ and τ be finite-state strategies for Player 0 respectively Player 1. Then, $\rho(s, \sigma, \tau)$ is ultimately periodic.*

2.9 Basic Results

To conclude this chapter, we state some results about the various games introduced so far, which are used in the later chapters. The most general one concerns Borel Games.

Theorem 2.13 ([37])**.** *Borel Games are determined.*

This result immediately implies the determinacy of all games introduced above (save Mean-Payoff Games, for which our notion of determinacy does not apply). However, pure determinacy is generally not enough. Positional or finite-state strategies suffice to win the games introduced above. A game is *positionally determined* if from every vertex one of the players has a positional winning strategy. Analogously, a game is *determined with finite-state strategies*, if from every vertex one of the players has a finite-state winning strategy. The existence of finite-state winning strategies is typically proven by a reduction to a simpler game. The following result is a cornerstone of the theory of infinite games.

Theorem 2.14 ([17]). *Parity Games are positionally determined.*

As Büchi Games are a special case of Parity Games, we obtain a similar result.

Corollary 2.15. *Büchi Games are positionally determined.*

Proof. The Büchi Game (G, F) is equivalent to the Parity Game over G with coloring c, where $c(s) = 2$ for $s \in F$ and $c(s) = 1$ for $s \notin F$ □

Determinacy of Muller Games can be derived most easily from the positional determinacy of Parity Games by a reduction, although it was first proven directly in [6]. The memory structure used in the reduction keeps record of the vertices, ordered by their latest visit in the play up to that position, equipped with a marker that signals the infinity set of a play. This structure, called *latest appearance record* (LAR), is an improvement of the *order vector*, introduced by McNaughton [38]. A formal exposition can be found in [25]. We just note that the size of the memory is bounded by $(|G|+1)!$.

Theorem 2.16 ([26]). *Muller Games are reducible to Parity Games with finite memory. Thus, they are determined with finite-state strategies.*

Another corollary completes the discussion about regular games started in Example 2.11.

Corollary 2.17. *Every regular game is determined and both players have finite-state winning strategies.*

Proof. Combine the construction from Example 2.11 and Theorem 2.16. □

The last type of zero-sum game we deal with here are LTL Games. They are a special case of the regular games discussed in Corollary 2.17, which is the key to the proof of the following theorem.

Theorem 2.18. LTL *Games are finite-state determined.*

We also note the complexity of solving LTL Games, as we use them as target for reductions.

Theorem 2.19 ([49, 45]). *Solving* LTL *Games is* **2EXPTIME***-complete. Solving solitary* LTL *Games is* **PSPACE***-complete.*

The complexity of solving games for several syntactic fragments of LTL is discussed in great detail by Alur and La Torre [3]. They show that the restriction to a subset of the operators can lower the complexity drastically.

Lastly, we consider Mean-Payoff Games. Since they are not zero-sum games, our notion of determinacy does not apply here. Instead, we say that a strategy σ for Player 0 *guarantees* $v \in \mathbb{R}$ for her if $v_0(\rho) \geq v$ for all ρ consistent with σ. Similarly, τ for Player 1 *guarantees* $v \in \mathbb{R}$ for him if $v_1(\rho) \leq v$ for all ρ consistent with τ.

Theorem 2.20 ([15, 55])**.** *Let* (G, s, d, l) *be an initialized Mean-Payoff Game. There exists a value* $v_M(\mathcal{G})$ *and positional strategies* σ *and* τ *such that* σ *and* τ *guarantee* $v_M(\mathcal{G})$ *for Player* 0 *respectively Player* 1*. Furthermore, the value and the strategies are effectively computable.*

Notice that the strategies σ and τ are optimal in the sense that there are no strategies that guarantee a better value for one of the players. Assume there is a strategy σ' for Player 0 that guarantees $v_0 > v_M(\mathcal{G})$. Then

$$v_M(\mathcal{G}) < v_0 = v_0(\rho(v, \sigma', \tau)) \leq v_1(\rho(s, \sigma', \tau)) \leq v_M(\mathcal{G}),$$

which is a contradiction. Analogously, there is no strategy τ' for Player 1 that guarantees $v_1 < v_M(\mathcal{G})$. Also, we have $v_0(\rho(s, \sigma, \tau)) = v_1(\rho(s, \sigma, \tau)) = v_M(\mathcal{G})$.

Chapter 3

Request-Response Games

Request-Response Games, first introduced by Wallmeier et. al. [54], are characterized by a very intuitive winning condition: some vertices of the arena are designated as requests while others are responses. Player 0's goal is to respond to every request. Formally, the winning condition of $\mathcal{G} = (G, (Q_j, P_j)_{j=1,\dots,k})$ consists of a finite collection of pairs (Q_j, P_j), where Q_j and P_j are subsets of the arena's vertices. We call the pair (Q_j, P_j) the *j-th (Request-Response) condition*. A *request (of condition j)* is a visit of a vertex in Q_j and a *response (of condition j)* is a visit of a vertex in P_j. Furthermore, a request of condition j is *open* after a finite play if there was a request of condition j that has not yet been responded. It is Player 0's goal to answer every request of condition j by a subsequent response, where a single response answers all open requests accepted so far. Formally, Player 0 wins a play ρ, iff

$$\forall j\, \forall n (\rho_n \in Q_j \rightarrow \exists n' \geq n : \rho_{n'} \in P_j).$$

If we label the arena such that $l(s) = \{q_j \mid s \in Q_j\} \cup \{p_j \mid s \in P_j\}$, then the set of winning plays is also specified by the LTL formula

$$\varphi := \bigwedge_{j=1}^{k} \mathbf{G} \left(q_j \rightarrow \mathbf{F} p_j \right).$$

Thus, $\mathcal{G}$ is equivalent to the LTL Game (G, φ). Conversely, every (generalized) Büchi-Game can easily be reduced to an equivalent Request-Response Game.

A classical example for controller synthesis is a busy intersection with traffic lights, equipped with sensors in the streets that detect cars waiting at a red light, and pedestrian lights with buttons for waiting pedestrians. The arena consists of vertices encoding the state of the system: the colors of the lights and flags for the requests. All undesirable (read: unsafe, i.e., too many green lights) states are ignored. The transitions model the changes of the color, sensor readings, and pushed buttons. If a light changes to green, its flag is set to false. The desired behavior, every request by a waiting car or pedestrian is granted, can be modeled by a Request-Response pair for every light. This example illustrates the need for time-optimal winning strategies. A traffic light that is green every other day satisfies the specification, but is not useful at all. Every request should be served as soon as possible. Nevertheless, it might be advantageous to prioritize some Request-Response conditions over others, for example if one of the streets is a major thoroughfare and the other is a small side street. All this can be implemented in the framework we introduce in this Chapter. However, the framework ignores multiple requests, thus the light with the most cars waiting is not served first. In Chapter 4 we discuss how to factor this in as well.

The intuitive notion of open requests naturally leads the way to the definition of waiting times: every time a condition is requested that is not open at the moment, a clock is started. This clock is stopped as soon as the request is responded. All requests of a condition that is already open at that moment are ignored. Thus, instead of just determining winning strategies, we are now interested in time-optimal strategies, i.e., strategies that minimize the waiting times. This changes the strategy problem from a decision problem to an optimization problem. Since a play is infinite, we need to aggregate the periods of waiting for a response to define the measure of a strategy. Then, the value of a strategy is the worst value of the plays consistent with the strategy.

A rather simple choice for aggregation is to uniformly bound the waiting times. Such a bound can be found by showing the existence of finite-state winning strategies: Request-Response Games are easily reducible to Büchi Games by keeping track of the open requests in an extra component of the expanded arena. The set F is chosen such that F is visited infinitely often iff no request is open indefinitely. This reduction does not only prove determinacy of Request-Response Games but also gives an upper bound on the waiting times: by playing according to the finite-state strategy derived from the reduction, Player 0 ensures that every request is open for at most $|G| \cdot k$ moves, provided that she has a winning strategy at all. However, this bound needs not to be optimal.

It might be desirable for Player 0 to keep one request open for more steps than she has to in order to satisfy other requests more quickly. This is especially true in cases where the conditions have different priorities, which can be modeled by penalty functions that aggravate the waiting times. Thus, the global bound on the waiting time might be very high, but the average waiting time decreases. This shows the need for an approach that aggregates the waiting times over the infinite duration of the play, thereby permitting a trade-off between the conditions. The average number of open request, the average waiting time, and the average accumulated waiting time are three types of aggregations discussed by Wallmeier [53]. He argues that only the latter one meets all desired properties of such a measurement: the winner of a game can be determined from its value and longer waiting times are increasingly penalized. Horn et. al. [29] showed that, with respect to the average accumulated waiting time, optimal finite-state strategies exist and can be computed. We will repeat this proof here, since it contains an error (in the proof of Proposition 9) which requires a modification of the proof technique. This is done in this chapter, in order to adapt the corrected technique to a novel winning condition presented in Chapter 4.

The proof consists of two steps. First, we show that it is not optimal to keep a request open arbitrarily long, but that there exists a bound such that waiting times above that bound are not worthwhile. The following observation is key: if a condition is open long enough, then the play visits a vertex twice such that the waiting times for all other conditions are higher at the second visit than they were at the first visit. Hence, Player 0 can play after the first visit as if it was the second visit. Thus, she skips a portion of the play without neglecting the other conditions. By skipping only costly loops, Player 0 can ensure that the value of the play decreases. Applying this infinitely often shows that an optimal strategy uniformly bounds the waiting times.

Thus, we can restrict our search for an optimal winning strategy to a finite domain. The second step of the proof consists of a straight-forward reduction from the problem of finding an optimal strategy for Request-Response Games to the same problem for Mean-Payoff Games. The memory of the expanded arena is used to keep track of the waiting times. The first step guarantees that this arena is still finite.

This chapter is structured as follows: we begin by reducing Request-Response Games to Büchi Games in Section 3.1, a result which has a corollary that turns out to be useful to us. In Section 3.2, we define waiting times and the value of a play and discuss some properties. Then, we are able to state the main theorem of this chapter and spend the rest of it to prove the theorem: in Subsection 3.2.1 we carry out the first step, showing that for every strategy of small value, there is another strategy of even smaller value that additionally bounds the waiting times of all conditions. The second step, the reduction to Mean-Payoff Games, is presented in Subsection 3.2.2. For the remainder of this chapter, let $G = (V, V_0, V_1, E)$ be an arena and let $\mathcal{G} = (G, s_0, (Q_j, P_j)_{j=1,\dots,k})$ be an initialized Request-Response Game.

3.1 Solving Request-Response Games

Request-Response Games are reducible to Büchi Games. This implies determinacy of $\mathcal{G}$ and the existence of finite-state winning strategies. We present the proof here, since it gives a bound on the value of an optimal strategy, which we present in Section 3.2, after we have given all necessary definitions.

Theorem 3.1 ([54]). *Request-Response Games are reducible to Büchi Games.*

Proof. The memory is used to keep track of the open requests. Furthermore, a counter is used to check that no condition is open indefinitely. Every time the counter changes its value a final state is visited. Therefore, we have to take precautions if there is only one condition: if $k = 1$, then we add another condition (Q_2, P_2) with $Q_2 = P_2 = \emptyset$. Let $\mathfrak{M} = (M, \text{init}, \text{update})$ where

- $M = 2^{[k]} \times [k] \times \{0, 1\}$,
- $\text{init}(s) = (\{j \in [k] \mid s \in Q_j \setminus P_j\}, 1, 0)$, and

- update$((M, m, f), s) = (M', m', f')$ where
  - $M' = (M \cup \{j \in [k] \mid s \in Q_j\}) \setminus \{j \in [k] \mid s \in P_j\}$,
  - $m' = \begin{cases} m & \text{if } m \in M' \\ (m \mod k) + 1 & \text{otherwise} \end{cases}$, and
  - $f' = \begin{cases} 0 & \text{if } m = m' \\ 1 & \text{otherwise} \end{cases}$.

To complete the definition of the Büchi Game, we specify the set $F = V \times 2^{[k]} \times [k] \times \{1\}$ of recurring states. So, we define $\mathcal{G}' = (G \times \mathfrak{M}, F)$ and have to show $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$. Therefore, let $\rho = \rho_0\rho_1\rho_2\ldots$ be a play in G and

$$\rho' = (\rho_0, (M_0, m_0, f_0))(\rho_1, (M_1, m_1, f_1))(\rho_2, (M_2, m_2, f_2))\ldots$$

be the unique expanded play in $G \times \mathfrak{M}$.

$$\begin{aligned} & \text{Player 1 wins } \rho \\ \Leftrightarrow\ & \exists j\, \exists n' : (\rho_n \in Q_j \land \forall n \geq n' : \rho_n \notin P_j) \\ \Leftrightarrow\ & \exists j\, \forall^\omega n : j \in M_n \\ \Leftrightarrow\ & \exists j\, \forall^\omega n : m_n = j \\ \Leftrightarrow\ & \forall^\omega n : f'_n = 0 \\ \Leftrightarrow\ & \forall^\omega n : \rho'_n \notin F \\ \Leftrightarrow\ & \text{Player 1 wins } \rho' \end{aligned}$$

□

This reduction is asymptotically optimal.

Lemma 3.2 ([54]). *There is a family of initialized Request-Response Games* $(\mathcal{G}_n)_{n\in\mathbb{N}}$ *such that*

(i) the size of the arena of $\mathcal{G}_n$ *is linear in* n*,*

(ii) the number of Request-Response conditions of $\mathcal{G}_n$ *is linear in* n*,*

(iii) Player 0 *wins* $\mathcal{G}_n$*, but*

(iv) she has no finite-state winning strategy of size less than $n \cdot 2^n$*.*

3.2 Time-optimal Strategies for Request-Response Games

In this section, we begin the treatment of time-optimal strategies for Request-Response Games by formalizing the intuitive notion of waiting times and by defining the value of a strategy, following [29]. The *waiting time* for condition j, $t_j : V^* \to \mathbb{N}$, is defined inductively by $t_j(\varepsilon) = 0$ and

- If $t_j(w) = 0$, then $t_j(ws) = \begin{cases} 1 & \text{if } s \in Q_j \setminus P_j \\ 0 & \text{otherwise} \end{cases}$
- If $t_j(w) > 0$, then $t_j(ws) = \begin{cases} 0 & \text{if } s \in P_j \\ t_j(w) + 1 & \text{otherwise} \end{cases}$.

Let $t(w) = (t_1(w), \dots, t_k(w))$ be the *waiting time vector*. We compare vectors componentwise, i.e., $t(x) \leq t(y)$ iff $t_j(x) \leq t_j(y)$ for all j. A strategy σ for Player 0 *uniformly bounds* the waiting time for condition j to B, if $t_j(w) \leq B$ for all finite plays w consistent with σ. From the definition of t_j we can directly derive the following remark about the evolution of the waiting time.

Remark 3.3. *$t_j(x) \leq t_j(y)$ implies $t_j(xs) \leq t_j(ys)$ for all $x, y \in V^*$ and all $s \in V$.*

The value of a play is determined by the average accumulated waiting time. However, we want to be able to prioritize some conditions. Thus, we use a penalty function f_j for every condition j to define the value of a play. We require f_j to be strictly increasing, since otherwise longer stretches of open requests could be desirable. Even worse, if we choose f_j to be a constant function, i.e., the penalty for an open request is the same, no matter how long the request is open already, then there are no optimal finite-state strategies as we show in Example 3.4.

So, given a family $(f_j : \mathbb{N} \to \mathbb{N})_{j=1,\dots,k}$ of strictly increasing *penalty functions*, define

- the *penalty for condition j after w*: $p_j(w) = f_j(t_j(w))$,
- the *penalty after w*: $p(w) = \sum_{j=1}^{k} p_j(w)$,
- the *value of a play* $v_R(\rho) = \limsup_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} p(\rho_0 \dots \rho_i)$, and
- the *value of a strategy* $v_R(\sigma) = \sup_{\tau \in \Gamma_1} v_R(\rho(s_0, \sigma, \tau))$.

A strategy σ for Player 0 is *optimal*, if $v_R(\sigma) \leq v_R(\sigma')$ for all $\sigma' \in \Gamma_0$.

Example 3.4. Consider the arena G with initial vertex s_0 given in Figure 3.1 with the winning condition comprised of the two conditions $(\{s_0\}, \{s_1\})$ and $(\{s_0\}, \{s_4\})$ and let the value of a play be defined with respect to the constant penalty functions f_1, f_2 with $f_1(t) = f_2(t) = 1$ for all $t > 0$ and $f_1(0) = f_2(0) = 0$.

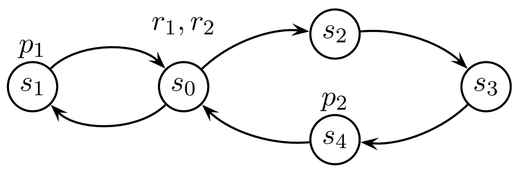

Figure 3.1: The arena G for a game that has no optimal winning strategies for constant penalty functions

Note that every winning strategy has to use both loops infinitely often. However, using the left loop twice instead of the right loop once in a play $\rho_0 \dots \rho_{n-1}$ decreases $\sum_{i=0}^{n-1} \sum_{j=1}^{2} f_j(t_j(\rho_0 \dots \rho_i))$ by one. Thus, for a given finite-state strategy σ, the unique resulting play $\rho(s, \sigma) = xy^\omega$ is ultimately periodic, by Remark 2.12. The period y visits the right loop at least once. Also, since G is a solitary arena, we have $v_R(\rho) = v_R(\sigma)$.

Let y' result from y by replacing every visit of the right loop by two visits of the left loop and let $\rho' = x(yy')^\omega$. This play is realized by a finite-state strategy σ'. Then, $v_R(\rho') < v_R(\rho)$ and therefore $v_R(\sigma') = v_R(\rho') < v_R(\rho) = v_R(\sigma)$. Thus, Player 0 has no optimal finite-state strategy.

We continue by some simple, but useful, observations about the value of a play respectively a strategy.

Lemma 3.5. *Let ρ be a play and σ a strategy for Player 0.*

(i) If $v_R(\rho) < \infty$, then Player 0 wins ρ.

(ii) If $v_R(\sigma) < \infty$, then σ is a winning strategy for Player 0.

Proof. (i) Consider the contraposition: let $\rho = \rho_0\rho_1\rho_2 \dots$ be winning for Player 1. Then, there exists a condition j that is requested at some ρ_n, but never responded at any $\rho_{n+n'}$. Then

$$t_j(\rho_0 \dots \rho_{n+n'}) = n' \leq f_j(t_j(\rho_0 \dots \rho_{n+n'})) \leq p(\rho_0 \dots \rho_{n+n'}),$$

since f_j is strictly increasing. Thus,

$$\frac{1}{n+n'} \sum_{i=0}^{n+n'-1} p(\rho_0 \dots \rho_i) \geq \frac{1}{n+n'} \cdot \frac{n' \cdot (n'-1)}{2} = \frac{n'-1}{2 \cdot \left(\frac{n}{n'} + 1\right)}$$

for all $n' \in \mathbb{N}$, which diverges to infinity. Therefore, $v_R(\rho) = \infty$.

(ii) Towards a contradiction assume σ is not a winning strategy. Then, there exists a strategy τ for Player 1 such that the play $\rho(s_0, \sigma, \tau)$ is won by Player 1. Then, $v_R(\rho(s_0, \sigma, \tau)) = \infty$ by (i) and therefore $v_R(\sigma) = \infty$, which yields the desired contradiction. □

The other implication of the statements does not hold: Player 0 can win a play, even if its value diverges. An example is a play such that the waiting time for a condition is not uniformly bounded, but every request is responded eventually. However, we show that this is not necessary. If Player 0 wins a game, then she can also win and uniformly bound the waiting times.

Now, we return to the reduction of Request-Response Games to Büchi Games. The strategy induced by the reduction uniformly bounds the waiting time in terms of the size of the arena and the number of Request-Response conditions. This also bounds the value of an optimal strategy. Our interest in the proof of Theorem 3.1 stems from the following corollary.

Corollary 3.6. *If Player* 0 *wins* $\mathcal{G}$, *then she also has a winning strategy* σ *such that* $v_R(\sigma) \leq \sum_{j=1}^{k} f_j(|G| \cdot k) =: b_R(\mathcal{G})$.

Proof. Let Player 0 win $\mathcal{G}$. Then, she also has a positional winning strategy σ' from $(s_0, \mathrm{init}(s_0))$ in $\mathcal{G}'$ from Theorem 3.1. Consider a play ρ' in $G \times \mathfrak{M}\!\upharpoonright_{\sigma'}$ and assume that there is an infix w of ρ' of length greater than $|G|$ such that w does not contain a vertex from F. This implies the existence of a play, played according to σ' that visits F only finitely often, which contradicts the fact that σ' is a winning strategy. Hence, every infix of length greater than $|G|$ of a play ρ that is consistent with σ' visits F at least once. This means that the m-component of the memory state of ρ' changes after at most $|G|$ moves. This component ranges over $[k]$; hence, after at most $|G| \cdot k$ moves, the m-component has cycled through all possible values. Thus, every j cannot be in the M-component of more than $|G| \cdot k$ consecutive memory states. The index j leaves the M-component at a vertex (s', m') of ρ' if $s' \in P_j$.

Now, consider a play $\rho = \rho_0\rho_0\rho_2 \dots$ of $\mathcal{G}$ according to the winning strategy σ obtained by the reduction. It is a projection of a play according to σ', which directly implies $t_j(\rho_0 \dots \rho_n) \leq |G| \cdot k$ for all n and all $j \in [k]$. Thus,

$$\begin{aligned}
\frac{1}{n}\sum_{i=0}^{n-1} p(\rho_0 \dots \rho_i) &= \frac{1}{n}\sum_{i=0}^{n-1}\sum_{j=1}^{k} f_j(t_j(\rho_0 \dots \rho_i)) \\
&\leq \frac{1}{n}\sum_{i=0}^{n-1}\sum_{j=1}^{k} f_j(|G| \cdot k) \\
&= \sum_{j=1}^{k} f_j(|G| \cdot k) = b_R(\mathcal{G}).
\end{aligned}$$

This implies $v_R(\rho) \leq b_R(\mathcal{G})$ for every play ρ consistent with σ. Therefore, we obtain $v_R(\sigma) \leq b_R(\mathcal{G})$. □

This completes our preparation and we are now able to state the main theorem of this chapter; The rest of this chapter is devoted to prove it. This result was already claimed in [29]. We present a new proof here, which can be adapted to other winning conditions. Nevertheless, the definition of the strategy improvement operator and the results about it are from [29]. Then, we deviate and define an improvement scheme transforming a winning strategy into a winning strategy that additionally bounds all waiting times, without increasing the value of the strategy.

Theorem 3.7. *If Player 0 wins $\mathcal{G}$, then she also has an optimal winning strategy. Furthermore, this strategy is finite-state and effectively computable.*

The proof consists of two major steps. The first one, presented in Subsection 3.2.1, is to prove that from every winning strategy of value less than $b_R(\mathcal{G})$, we can construct another winning strategy with smaller value that uniformly bounds the waiting times for all conditions. This is achieved by improving the strategy repeatedly with a strategy improvement operator that deletes costly loops of the plays consistent with the strategy under consideration. Hence, if Player 0 wins $\mathcal{G}$, then $v_R(\sigma) \leq b_R(\mathcal{G})$ for the optimal winning strategy σ by Theorem 3.1 and Corollary 3.6. Thus, an optimal strategy also bounds the waiting times for all conditions. These bounds allow us to find an optimal strategy in the second step by reducing the Request-Response Game to a Mean-Payoff Game, presented in Subsection 3.2.2. The expanded game is constructed such that the plays of both games have the same value. Thus, an optimal positional strategy for the Mean-Payoff Game, which exists by Theorem 2.20, induces an optimal strategy for the Request-Response Game.

3.2.1 Strategy Improvement for Request-Response Games

In this subsection, we prove the following statement: if Player 0 has a winning strategy of value less than $b_R(\mathcal{G})$, then she also has a strategy of smaller or equal value, which uniformly bounds the waiting times of all conditions. Let σ_0 be a winning strategy for Player 0 with value $v_R(\sigma_0) \leq b_R(\mathcal{G})$. Inductively, we define strategies $\sigma_1, \dots, \sigma_k$ with the following properties.

- All strategies are winning strategies for Player 0,
- $b_R(\mathcal{G}) \geq v_R(\sigma_0) \geq v_R(\sigma_1) \geq \cdots \geq v_R(\sigma_k)$, and
- σ_j uniformly bounds the waiting times for all conditions $j' \leq j$ to some bound that only depends on the size of the arena and the number of Request-Response conditions.

The first part of this subsection repeats work of Horn et. al. [29]. A strategy improvement operator is defined for every condition and we show that the value does not increase, if the strategy is improved. In the second part of this subsection, which contains novel material, we define an improvement scheme which applies each strategy improvement operator infinitely often to a given winning strategy σ_{j-1}, obtaining the limit strategy σ_j. Here, our proof differs from the work in [29], which applies the strategy improvement operator for condition only once. Afterwards, we have to lift the properties of a single improvement step to the limit of the improved strategies. Then, we are able to prove that the limit strategies do bound the waiting times.

Intuitively, the strategy σ_j from above is defined as σ_{j-1} unless the waiting time for condition j exceeds some bound. In this case, Player 0 skips loops in which the request is not responded. However, she also has to make sure that she does not miss a response of some other condition that might be open from that point onwards. Therefore, she only skips loops in which the waiting time of all conditions at the end of the loop is greater than the waiting time at the beginning. By deleting loops, she might form new loops that could be deleted, also. Therefore, the strategy improvement operator has to be applied infinitely often. If there are no more loops to skip, then the length of an infix, in which a request is open continuously, can be bounded. Mathematically speaking, we apply the strategy improvement operator infinitely often and define σ_j to be the limit of the improved strategies. In the following, we introduce the *strategy improvement operator* I_j for the j-th condition and discuss some useful properties as presented in [29]. These properties are then lifted to the limit of the improved strategies.

Given a winning strategy σ such that $v_R(\sigma) \leq b_R(\mathcal{G})$, we define the improved strategy $I_j(\sigma)$ as a strategy with memory $\mathfrak{M} = (M, \text{init}, \text{update})$ and next-move function next as follows: The set of memory states M consists of finite plays played according to σ and is defined implicitly. The initialization function is given by $\text{init}(s_0) = s_0$. The update function is going to be defined such that the last vertex of w and $\text{update}^*(w)$ are equal. So, when defining $\text{update}(w, s)$ for $w \in M$ and $s \in V$, we can assume that the last vertex of w and s are connected by an edge in G. We consider two cases.

If condition j is not even open or Player 0 has not waited too long, she continues to play according to σ. Therefore, we define $\text{update}(w, s) = ws$ if $t_j(ws) \leq f_j^{-1}(b_R(\mathcal{G}))$. If she has waited too long, she looks ahead to skip a loop that she would have played according to σ, but which does not contain a response of condition j. However, she has to make sure that she does not carelessly skip responses of the other conditions if they are open. Thus, if $t_j(ws) > f_j^{-1}(b_R(\mathcal{G}))$, consider the tree obtained from $\mathfrak{T}^{\sigma}_{G,s_0}\!\upharpoonright_{ws}$ by deleting on every path the subtrees attached to the first vertex belonging to P_j. This tree is finite, since every infinite path corresponds to a losing play for Player 0 played according to σ. This cannot happen, since σ is a winning strategy. Now consider the set of vertices z of the tree, such that z ends in s and $t(ws) \leq t(z)$. This set is non-empty as it contains ws. Let $z's$ be a vertex of maximal depth in this set. Then, $\text{update}(w, s) = z's$.

To finish the definition of the strategy, we define $\text{next}(s, w) = \sigma(w)$. It is clear that the last vertices of w and $\text{update}^*(w)$ are equal for every prefix w consistent with $I_j(\sigma)$, and that I_j behaves as intended, i.e., w is a subword of $\text{update}^*(w)$ for every w consistent with $I_j(\sigma)$. Also, update^* is monotonous with respect to the prefix relation, i.e., if $x \sqsubseteq y$, then also $\text{update}^*(x) \sqsubseteq \text{update}^*(y)$. Furthermore, in both cases of the definition, the updated memory is a prefix of a play according to σ.

Lemma 3.8 ([29]). $\text{update}^*(w)$ *is consistent with* σ *for all* w *consistent with* $I_j(\sigma)$.

Proof. By induction over w: every play starts in s_0, so the induction base is trivial, as $\text{update}^*(s_0) = \text{init}(s_0) = s_0$ holds. For the induction step, let $w = \rho_0 \dots \rho_n$ be played according to $I_j(\sigma)$. Applying the induction hypothesis, we can assume that $\text{update}^*(\rho_0 \dots \rho_{n-1})$ is consistent with σ. By definition of update we have

$$\text{update}^*(\rho_0 \dots \rho_n) = \text{update}(\text{update}^*(\rho_0 \dots \rho_{n-1}), \rho_n).$$

Furthermore, as noted above, the last vertex of $\text{update}^*(\rho_0 \dots \rho_{n-1})$ is ρ_{n-1}. Finally, if $\rho_{n-1} \in V_0$, then

$$\begin{aligned}\rho_n &= I_j(\sigma)(\rho_0 \dots \rho_{n-1}) \\ &= \text{next}(\rho_{n-1}, \text{update}^*(\rho_0 \dots \rho_{n-1})) \\ &= \sigma(\text{update}^*(\rho_0 \dots \rho_{n-1})).\end{aligned} \tag{3.1}$$

Analogously to the definition, we consider two cases: either, we have

$$\text{update}(\text{update}^*(\rho_0 \dots \rho_{n-1}), \rho_n) = \text{update}^*(\rho_0 \dots \rho_{n-1})\rho_n.$$

Then, by induction hypothesis and (3.1), $\text{update}^*(\rho_0 \dots \rho_n)$ is consistent with σ_{j-1}. Otherwise, in the second case of the definition, we have

$$\text{update}(\text{update}^*(\rho_0 \dots \rho_{n-1}), \rho_n) = \text{update}^*(\rho_0 \dots \rho_{n-1})\rho_n z',$$

where z' is a path in $\mathfrak{T}^{\sigma}_{G,s_0}\upharpoonright_{\text{update}^*(\rho_0 \dots \rho_{n-1})\rho_n}$. Together with the induction hypothesis and (3.1), this shows that $\text{update}^*(\rho_0 \dots \rho_n)$ is consistent with σ. □

This allows us to bound the waiting time of a finite play by the waiting time of its memory content.

Lemma 3.9 ([29])**.** *$t(w) \leq t(\text{update}^*(w))$ for all w consistent with $I_j(\sigma)$.*

Proof. By induction over w. The base case is $t(s_0) = t(\text{init}(s_0)) = t(\text{update}^*(s_0))$. For the induction step we can assume $t(w) \leq t(\text{update}^*(w))$. By Remark 3.3, we get $t(ws) \leq t(\text{update}^*(w)s)$. Now consider two cases: if $t_j(\text{update}^*(w)s) \leq f_j^{-1}(b_R(\mathcal{G}))$, we have

$$\text{update}^*(ws) = \text{update}(\text{update}^*(w), s) = \text{update}^*(w)s$$

and thus

$$t(ws) \leq t(\text{update}^*(w)s) = t(\text{update}^*(ws)).$$

On the other hand, if $t_j(\text{update}^*(w)s) > f_j^{-1}(b_R(\mathcal{G}))$, we have

$$\text{update}^*(ws) = \text{update}(\text{update}^*(w), s) = zs$$

where $t(zs) \geq t(\text{update}^*(w)s)$ and thus

$$t(ws) \leq t(\text{update}^*(w)s) \leq t(zs) = t(\text{update}^*(ws)).$$ □

This result implies that the uniform bounds of σ are also realized by $I_j(\sigma)$.

Lemma 3.10 ([29]). *If σ uniformly bounds the waiting time for some condition j' to B, so does $I_j(\sigma)$.*

Proof. We have $t_{j'}(w) \leq B$ for all finite plays w consistent with σ. Now, let w be a play consistent with $I_j(\sigma)$. Since update$^*(w)$ is played according to σ we conclude $t_{j'}(\text{update}^*(w)) \leq B$ and $t_{j'}(w) \leq t_{j'}(\text{update}^*(w))$ by Lemma 3.9. Thus, $I_j(\sigma)$ uniformly bounds the waiting time for condition j' to B. ☐

Now, we show that the value of the improved strategy does not increase, applying Lemma 3.9.

Lemma 3.11 ([29]). *$v_R(I_j(\sigma)) \leq v_R(\sigma)$.*

Proof. For $\rho = \rho_0\rho_1\rho_2\dots$ let

$$\text{update}^*(\rho) = \lim_{n\to\infty} \text{update}^*(\rho_0 \dots \rho_n),$$

which is consistent with σ for every play ρ consistent with $I_j(\sigma)$ by Lemma 3.8. We show $v_R(\rho) \leq v_R(\text{update}^*(\rho))$ for all ρ consistent with $I_j(\sigma)$, which implies the claim. To this end, we define

$$S = \{w' \sqsubset \text{update}^*(\rho) \mid \neg\exists w \sqsubset \rho : \text{update}^*(w) = w'\}.$$

S contains exactly the vertices of the loops skipped by Player 0. Let $w' \in S$: by definition of update, we know $t_j(w') > f_j^{-1}(b_R(\mathcal{G}))$ and thus $p(w') > b_R(\mathcal{G}) \geq v_R(\sigma)$. Hence,

$$\limsup_{n\to\infty} \frac{1}{n}\sum_{i=0}^{n-1} p(\text{update}^*(\rho_0 \dots \rho_i)) \leq v_R(\text{update}^*(\rho)) \tag{3.2}$$

since the average decreases, if the summation omits the summands for the prefixes in S. Now, let $w \sqsubset \rho$: we have $t(w) \leq t(\text{update}^*(w))$ by Lemma 3.9 and therefore $p(w) \leq p(\text{update}^*(w))$. Thus,

$$\frac{1}{n}\sum_{i=0}^{n-1} p(\rho_0 \dots \rho_i) \leq \frac{1}{n}\sum_{i=0}^{n-1} p(\text{update}^*(\rho_0 \dots \rho_i)).$$

The latter term converges to a value less than or equal to $v_R(\text{update}^*(\rho))$ by (3.2). Thus, we conclude $v_R(\rho) \leq v_R(\text{update}^*(\rho))$. ☐

We can immediately conclude that σ_j is a winning strategy for Player 0.

Corollary 3.12 ([29]). *$I_j(\sigma)$ is a winning strategy for Player 0, if σ is a winning strategy.*

Proof. We have $b_R(\mathcal{G}) \geq v_R(\sigma) \geq v_R(I_j(\sigma))$ by Lemma 3.11. Applying Lemma 3.5 proves the statement. □

This concludes our discussion of the strategy improvement operator I_j. Proposition 9 of [29] claims that the waiting times for all conditions are already bounded, if each I_j is applied only once. However, the proof is erroneous. To overcome this, the remainder of this subsection is used to present a modified proof, which applies every I_j infinitely often. We prove that the limit of these strategies exists and bounds the waiting time for condition j. Additionally, we have to lift the lemmata stated above for I_j to the limit of the improved strategies. Let σ_0 be a winning strategy such that $v_R(\sigma_0) \leq b_R(\mathcal{G})$. We define

- $\sigma_{j,0} = \sigma_{j-1}$ for $j \in [k]$,
- $\sigma_{j,n+1} = I_j(\sigma_{j,n})$ for $j \in [k]$ and $n \geq 0$, and
- $\sigma_j = \lim_{n\to\infty} \sigma_{j,n}$ for $j \in [k]$.

The improvement scheme is visualized in Figure 3.2. Before we begin to discuss the scheme, we have to show that it is well-defined, i.e., $(\sigma_{j,n})_{n\in\mathbb{N}}$ converges. To this end, let $(M_{j,n}, \mathrm{init}_{j,n}, \mathrm{update}_{j,n})$ be the memory structure and $\mathrm{next}_{j,n}$ be the next-move function used to define $\sigma_{j,n}$.

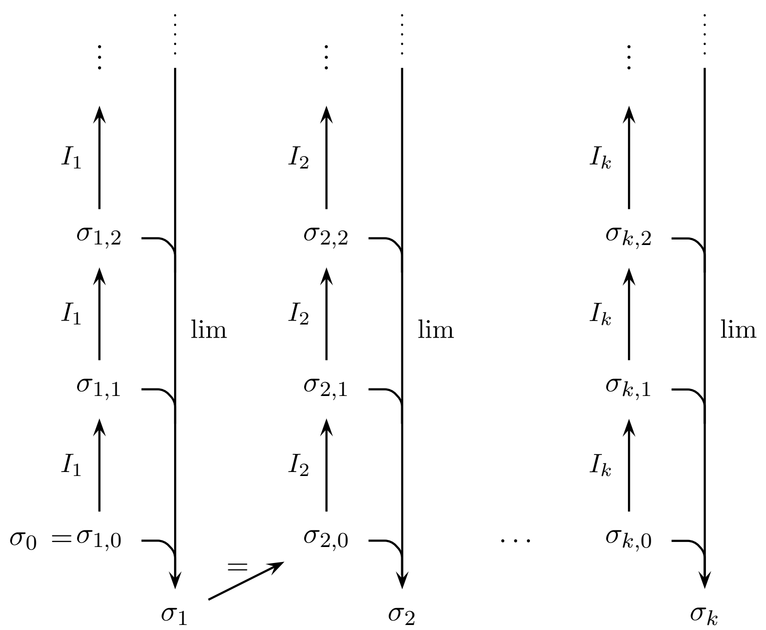

Figure 3.2: The improvement scheme for a Request-Response Game

Lemma 3.13. *Let $j \in [k]$. Then,* $\lim_{n\to\infty} \sigma_{j,n}$ *exists.*

Proof. First, we show by induction that $(\text{update}^*_{j,n})_{n\in\mathbb{N}}$ converges to the identity function: for the induction base, we have $\text{update}^*_{j,n}(s_0) = s_0$ for all n. Now assume that $\text{update}^*_{j,n}(w) = w$ for all $n \geq n_w$. If $t_j(ws) \leq f_j^{-1}(b_R(\mathcal{G}))$, then $\text{update}_{j,n}(w,s) = ws$. Hence, $\text{update}^*_{j,n}(ws) = ws$ for all $n \geq n_w = n_{ws}$. Thus, let $t_j(ws) > f_j^{-1}(b_R(\mathcal{G}))$ and let $\mathfrak{T}_n$ be $\mathfrak{T}^{\sigma_{j,n-1}}_{G,s_0}\lceil_{ws}$ limited to the first visit of P_j. By definition, $\text{update}^*_{j,n}(ws)$ is a vertex of $\mathfrak{T}_n$. Since every path in $\mathfrak{T}_n$ is a path in $\mathfrak{T}_{n-1}$ from which some loops might be deleted, the size of the $\mathfrak{T}_n$ is decreasing. Finally, if $\mathfrak{T}_n = \mathfrak{T}_{n+1}$, then we have $\mathfrak{T}_{n'} = \mathfrak{T}_n$ for all $n' \geq n$. Thus, there is an index $n_{ws} \geq n_w$ such that the $\mathfrak{T}_n$ are equal for all $n \geq n_{ws}$. From that index on, we have $\text{update}_{j,n}(w,s) = ws$ and thus $\text{update}^*_{j,n}(ws) = ws$. Thus, the sequence of the update functions converges to the identity function.

Now, we have

$$\begin{aligned}
&\sigma_{j,n}(\rho_0 \dots \rho_i)\\
=&\text{next}_{j,n}(\rho_i, \text{update}^*_{j,n}(\rho_0 \dots \rho_i))\\
=&\text{next}_{j,n}(\rho_i, \rho_0 \dots \rho_i)\\
=&\sigma_{j,n-1}(\rho_0 \dots \rho_i)
\end{aligned}$$

for all plays $\rho_0 \dots \rho_i$ and all sufficiently large n. Thus, $(\sigma_{j,n})_{n\in\mathbb{N}}$ converges. □

Remark 3.14. *$x \sqsubseteq y$ implies $n_x \leq n_y$ for all x and y.*

To complete the discussion we lift the results stated in the Lemmata 3.10 and 3.11 for a single improvement step to the limit of the infinitely many improvement steps. Therefore, we need to compose the functions $\text{update}^*_{j,n}$ in a backwards manner in order to determine for every play ρ' consistent with σ_j the play ρ consistent with σ_{j-1} such that ρ' is obtained from ρ by skipping some loops. This can be done, since $(\sigma_{j,n})_{n\in\mathbb{N}}$ converges. So, every finite play w that is consistent with σ_j, is consistent with almost all $\sigma_{j,n}$. To this end, we define $\text{update}^*_{j,[m,n]}$ for $1 \leq m \leq n$ by

$$\text{update}^*_{j,[m,m]}(w) = \text{update}^*_{j,m}(w)$$

and

$$\text{update}^*_{j,[m,n+1]}(w) = \text{update}^*_{j,[m,n]}(\text{update}^*_{j,n+1}(w)).$$

Example 3.15. This construction is illustrated in Figure 3.3, where a request of j is open in the dashed intervals. Assume x is consistent with $\sigma_{3,n}$. Then, we have $x_2 = \text{update}^*_{j,3}(x) = \text{update}^*_{j,[3,3]}(x)$, which is consistent with $\sigma_{j,2}$. Applying $\text{update}^*_{j,2}$, we obtain $x_1 = \text{update}^*_{j,2}(x_2) = \text{update}^*_{j,[2,3]}(x)$, which is consistent with $\sigma_{j,1}$. Finally, we apply $\text{update}^*_{j,1}$, and obtain $x_0 = \text{update}^*_{j,1}(x_1) = \text{update}^*_{j,[1,3]}(x)$, which is consistent with $\sigma_{j,0} = \sigma_{j-1}$.

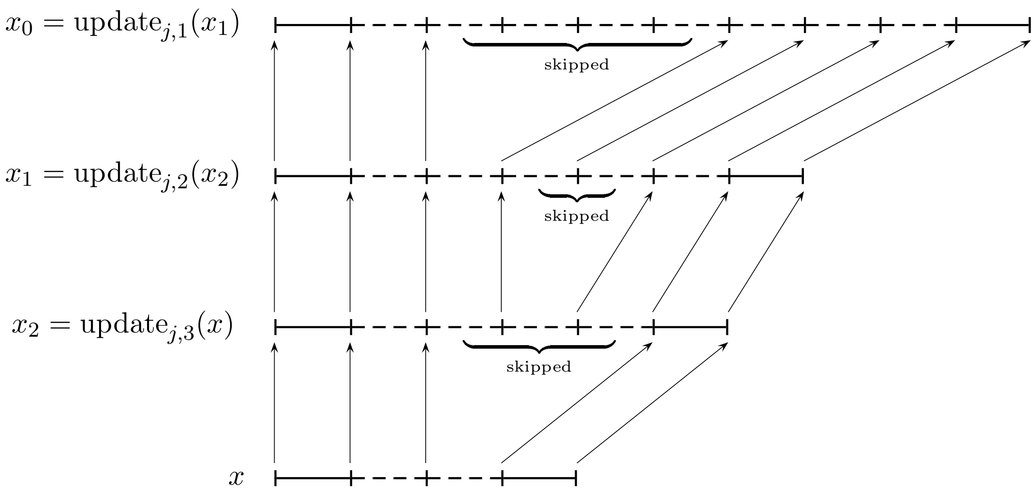

Figure 3.3: Reconstruction of a play with $\text{update}^*_{j,[m,n]}$

This reconstruction can be generalized. Applying Lemma 3.8 inductively, we can show that $\text{update}^*_{j,[m,n]}(w)$ is a finite play consistent with $\sigma_{j,m-1}$ for every play w consistent with $\sigma_{j,n}$. Especially $\text{update}^*_{j,[1,n_w]}(w)$ is consistent with $\sigma_{j,0} = \sigma_{j-1}$. Analogously, applying Lemma 3.9 inductively, we get $t(w) \leq t(\text{update}^*_{j,[m,n]}(w))$ for all w consistent with σ_n.

As $(\sigma_{j,n})_{n\in\mathbb{N}}$ converges, we are able to reconstruct the original play for every play that is consistent with σ_j, i.e., with the limit of the $\sigma_{j,n}$. For every finite play x according to σ_j, there is an n_x such that $\text{update}^*_{j,n}(x) = x$ for all $n \geq n_x$. We define $\text{update}^*_{j,\omega}$ by

$$\text{update}_{j,\omega}(x) = \text{update}^*_{j,[1,n_x]}(x).$$

Example 3.16. Going back to Figure 3.3, assume $n_x \leq 3$, i.e., x is consistent with $\sigma_{j,n}$ for all $n \geq 3$ and especially σ_j. Then, we have $\text{update}^*_{j,\omega}(x) = x_0$, which is a play consistent with σ_{j-1}.

Again, by the remarks from above, we know that $\text{update}^*_{j,\omega}(x)$ is consistent with σ_{j-1} and $t(x) \leq t(\text{update}^*_{j,\omega}(x))$ for every play x consistent with σ_j. Now we are able to lift the results to the limit of the improvement steps.

Lemma 3.17. *Let $j \in [k]$.*

(i) If σ_{j-1} uniformly bounds the waiting time for condition j' to B, then so does σ_j.

(ii) $v_R(\sigma_j) \leq v_R(\sigma_{j-1})$.

Proof. Both proofs are analogous to the proofs of Lemma 3.10 and Lemma 3.11 with $\text{update}^*_{j,\omega}$ instead of update^*.

(i) We have $t_{j'}(x) \leq B$ for all finite plays x consistent with σ_{j-1}. Now, let x be a play consistent with σ_j. Since $\text{update}^*_{j,\omega}(x)$ is a prefix of a play according to σ_{j-1} we conclude $t_{j'}(x) \leq t_{j'}(\text{update}^*_{j,\omega}(x)) \leq B$. Thus, σ_j uniformly bounds the waiting time for condition j'.

(ii) For $\rho = \rho_0\rho_1\rho_2\dots$ let

$$\text{update}^*_{j,\omega}(\rho) = \lim_{n\to\infty} \text{update}^*_{j,\omega}(\rho_0\dots\rho_n),$$

which is consistent with σ_{j-1} for every play ρ consistent with σ_j. Again, we show $v_R(\rho) \leq v_R(\text{update}^*_{j,\omega}(\rho))$ for all ρ consistent with σ_j, which implies the claim. To this end, we define

$$S = \{x' \sqsubset \text{update}^*_{j,\omega}(\rho) \mid \neg\exists x \sqsubset \rho : \text{update}^*_{j,\omega}(x) = x'\}.$$

Here, S contains exactly the vertices of the loops skipped by Player 0 throughout all improvement steps. Let $x' \in S$, then $t_j(x') > f_j^{-1}(b_R(\mathcal{G}))$ still holds, as every improvement step only deletes loops of $\text{update}^*_{j,\omega}(\rho)$ after a waiting time of at least $f_j^{-1}(b_R(\mathcal{G}))$ steps. Thus, $p(x') > b_R(\mathcal{G}) \geq v_R(\sigma)$ and

$$\limsup_{n\to\infty} \frac{1}{n}\sum_{i=0}^{n-1} p(\text{update}^*(\rho_0\dots\rho_i)) \leq v_R(\text{update}^*(\rho)) \tag{3.3}$$

since the average decreases, if the summation omits the summands for the prefixes in S.

Lastly, let $x \sqsubset \rho$. We have $t(x) \leq t(\text{update}^*_{j,\omega}(x))$ by the above remarks and therefore $p(x) \leq p(\text{update}^*_{j,\omega}(x))$. Thus,

$$\frac{1}{n}\sum_{i=0}^{n-1} p(\rho_0\dots\rho_i) \leq \frac{1}{n}\sum_{i=0}^{n-1} p(\text{update}^*(\rho_0\dots\rho_i)).$$

The latter term converges to a value less than or equal to $v_R(\text{update}^*(\rho))$ by (3.3). Thus, we conclude $v_R(\rho) \leq v_R(\text{update}^*(\rho))$. □

Again, we obtain the following corollary.

Corollary 3.18. *σ_j is a winning strategy for Player 0 for all $j \in [k]$.*

Proof. We have $b_R(\mathcal{G}) \geq v_R(\sigma_0) \geq v_R(\sigma_1)\dots \geq v_R(\sigma_j)$ by Lemma 3.17. Applying Lemma 3.5 yields the result. □

It remains to show that σ_j uniformly bounds the waiting time for condition j. A sequence $t_1,\dots,t_n \in \mathbb{N}^k$ of vectors is called *non-dickson*, if there is no pair of indices $1 \leq i < j \leq n$ such that $t_i \leq t_j$. If the vectors are waiting time vectors of a play such that condition j is open continuously, then the strategy improvement operator I_j does not skip a loop in that period. Dickson's Lemma [12] guarantees that such a sequence cannot be infinite, but this does not suffice for our purpose. Since the entries in our vectors can only increase by one, be reset to zero or stay zero, we are able to obtain a finite bound that only depends on the number of conditions and the size of the arena.

Let w_1 and w_2 be finite plays such that $w_1 \sqsubset w_2$. We say that the prefixes w_1 and w_2 are *dickson save condition* j if the last vertices of w_1 and w_2 are equal and $t_{j'}(w_1) \leq t_{j'}(w_2)$ for all $j' \neq j$. Now, let xy be a play. We say that y is *non-dickson save condition* j if xy does not have a pair of dickson prefixes that both contain a non-empty part of y. If y is non-dickson save condition j, then there is no loop in y that is deleted by the strategy improvement operator I_j. We bound the length of such a sequence in terms of the size of the arena and the number of conditions by a function b, following [53].

If there is only one condition, namely j, there must not be a state repetition in y. Thus, the length of a non-dickson sequence can be bounded by the number of vertices in $|G|$; hence, we define $b(m, 1) := m$. Now assume that there are $k + 1$ conditions. Every condition $j' \neq j$ has to be requested and responded at least once in every infix of length $b(|G|, k)$: assume condition j' does not. Then its waiting time is non-decreasing throughout the infix and two dickson prefixes for the other k conditions that exist by assumption, are also dickson for $k + 1$ conditions, which is what we try to avoid. Thus, the waiting times are bounded by $b(|G|, k)$ after an initial prefix of length $b(|G|, k)$. Thus, there cannot be more than

$$b(|G|, k) + |G| \cdot b(|G|, k)^k$$

steps without two dickson prefixes. Thus,

$$b(m, k + 1) := b(m, k) + m \cdot b(m, k)^k.$$

Lemma 3.19. *σ_j uniformly bounds the waiting time for condition j to $f_j^{-1}(b_R(\mathcal{G})) + b(|G|, k)$.*

Proof. Towards a contradiction, assume that there is a play w consistent with σ_j such that $t_j(w) > f_j^{-1}(b_R(\mathcal{G})) + b(|G|, k)$ and let $w = xy$ such that $|y| = b(|G|, k) + 1$. Then, $t_j(xy') > f_j^{-1}(b_R(\mathcal{G}))$ for all $y' \sqsubseteq y$, $y' \neq \varepsilon$. Furthermore, there exists an N such that $\text{update}^*_{j,N}(xy') = xy'$ for all $y' \sqsubseteq y$. Thus, y is a non-dickson sequence since there cannot be two dickson prefixes in y, as they would be skipped by I_j at some stage n. But this yields the desired contradiction as $|y| = b(|G|, k) + 1$ contradicts the definition of b. □

The preceding results are now combined to prove the existence of a winning strategy with value less than or equal to the original strategy's value that additionally uniformly bounds all waiting times.

Lemma 3.20. *For every winning strategy σ_0 for $\mathcal{G}$ for Player 0 of value $v_R(\sigma_0) \leq b_R(\mathcal{G})$, there is a winning strategy σ_k that uniformly bounds the waiting time for all conditions $j \in [k]$ to $f_j^{-1}(b_R(\mathcal{G})) + b(|G|, k)$. Furthermore, $v_R(\sigma_k) \leq v_R(\sigma_0)$.*

Proof. By induction over j. The strategy σ_j is a winning strategy by Corollary 3.12 and by induction hypothesis. It uniformly bounds the waiting time for condition j to $f_j^{-1}(b_R(\mathcal{G})) + b(|G|, k)$ by Lemma 3.19 and the waiting time for the conditions $j' < j$ to $f_{j'}^{-1}(b_R(\mathcal{G})) + b(|G|, k)$ by induction hypothesis and by Lemma 3.17. Furthermore, $v_R(\sigma_j) \leq v_R(\sigma_{j-1}) \leq \sigma_0$ by Lemma 3.17 and by induction hypothesis. □

This concludes the first step: we have proved that the search for an optimal strategy can be restricted to a finite domain.

3.2.2 Reducing Request-Response Games to Mean-Payoff Games

This subsection concludes the proof of Theorem 3.7 by constructing a Mean-Payoff Game whose plays are the plays of $\mathcal{G}$ annotated with their (bounded) waiting times. This construction is from [29], while the correctness proof is our own work. Lemma 3.20 guarantees that the waiting times can be bounded by

$$b_j := f_j^{-1}(b_R(\mathcal{G})) + b(|G|, k).$$

So, Player 0 can play optimally without ever keeping a request open for more than b_j steps. If a waiting time grows too big, then the play in the expanded arena ends up in a sink component. Thus, the expanded arena is finite. To construct it, we define the memory structure $\mathfrak{M} = (M, \text{init}, \text{update})$ where

- $M = \left(\prod_{j=1}^{k}\{0, \ldots, b_j\}\right) \cup \{m_\uparrow\}$,
- $\text{init}(s) = (t_1, \ldots, t_k)$ where $t_j = 1$ if $s \in Q_j \backslash P_j$ and $t_j = 0$ otherwise, and
- update is given by
  - $\text{update}(m_\uparrow, s) = m_\uparrow$ for all vertices s,
  - if $(t_1, \ldots, t_k)$ such that $t_j = b_j$ and $s \notin P_j$ for some j, then $\text{update}((t_1, \ldots, t_k), s) = m_\uparrow$, and
  - otherwise $\text{update}((t_1, \ldots, t_k), s) = (t'_1, \ldots, t'_k)$ where
$$t'_j = \begin{cases} 0 & \text{if } t_j = 0 \text{ and } s \notin Q_j \backslash P_j \\ 1 & \text{if } t_j = 0 \text{ and } s \in Q_j \backslash P_j \\ 0 & \text{if } t_j > 0 \text{ and } s \in P_j \\ t_j + 1 & \text{if } t_j > 0 \text{ and } s \notin P_j \end{cases}.$$

Since in a Mean-Payoff Game it is Player 1's goal to minimize the limit superior of the average edge weights, we have to switch the Player's positions. We define $V'_0 = V_1 \times M$, $V'_1 = V_0 \times M$ and $G' = (V \times M, V'_0, V'_1, E_{\text{update}})$. To complete the definition of the game, we need to give the weight function l: Let

$$d = \sum_{j=1}^{k} f_j(b_j + 1).$$

Then, $l((s,m),(s',m_\uparrow)) = d$ for all $(s,m) \in V \times M$ and

$$l((s,(t_1,\dots,t_k)),(s',(t'_1,\dots,t'_k))) = \sum_{j=1}^{k} f_j(t_j).$$

Now, we define the Mean-Payoff Game $\mathcal{G}' = (G', (s_0, \mathrm{init}(s_0)), d, l)$.

The following remark lists some simple observations about the connections between the plays of $\mathcal{G}$ and $\mathcal{G}'$.

Remark 3.21. *(i) Let* $\mathrm{update}(w) \neq m_\uparrow$*. Then,* $t(w) = \mathrm{update}^*(w)$*. Furthermore,* $\mathrm{update}^*(w) = m_\uparrow$ *iff there exists a prefix* $w' \sqsubseteq w$ *such that* $t_j(w') > b_j$ *for some* j*.*

(ii) Let ρ be a play in G such that the waiting time for every condition j is uniformly bounded by b_j and let ρ' be the expanded play in G'. Then, $v_R(\rho) = v_1(\rho')$.

(iii) Let ρ' be a play in G' that does not visit a vertex with memory state $m_\uparrow$ and ρ the projected play in G. Then, $v_1(\rho') = v_R(\rho)$.

Now, we are able to prove Theorem 3.7, which stated that Player 0 has an optimal winning strategy if she wins $\mathcal{G}$.

Proof. We begin by relating strategies and values for Player 0 for $\mathcal{G}$ and Player 1 in $\mathcal{G}'$.

Let σ be a strategy for Player 0 for $\mathcal{G}$ that uniformly bounds the waiting times for all j to b_j. We define τ' for Player 1 in $\mathcal{G}'$ by $\tau'((\rho_0,m_0)\dots(\rho_n,m_n)) = \sigma(\rho_0\dots\rho_n)$. We claim τ' guarantees $v_R(\sigma)$ for Player 1 in $\mathcal{G}'$. Assume it does not. Then, Player 0 has a strategy σ' for $\mathcal{G}'$ such that $v_1(\rho(s_0,\sigma',\tau')) > v_R(\sigma)$. The projected play ρ of $\rho(s_0,\sigma',\tau')$ is consistent with σ by construction of τ'. Thus, $v_R(\sigma) \geq v_R(\rho) = v_1(\rho(s_0,\sigma',\tau')) > v_R(\sigma)$ by Remark 3.21 (ii), which amounts to a contradiction.

Conversely, let τ' be a strategy for Player 1 in $\mathcal{G}'$ that guarantees a loss $d' < d$. Thus, no play consistent with τ' visits a vertex with memory state $m_\uparrow$. Let σ be the strategy for Player 0 for $\mathcal{G}$ induced by τ' via $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$. We claim $v_R(\sigma) \leq d'$. Assume Player 1 has a strategy τ for $\mathcal{G}$ such that $v_R(\rho(s_0,\sigma,\tau)) > d'$. The expanded play ρ' of $\rho(s_0,\sigma',\tau')$ is consistent with τ' by Lemma 2.8. Thus, $d' \geq v_1(\rho') = v_R(\rho(s_0,\sigma,\tau)) > d'$ by Remark 3.21 (iii), which again amounts to a contradiction.

Now, we can begin with the actual proof: since Player 0 wins $\mathcal{G}$, Corollary 3.6 and Lemma 3.20 guarantee that she also has a winning strategy σ that uniformly bounds the waiting times for all conditions j to b_j. Let τ' be the induced strategy for Player 1 in $\mathcal{G}'$. Every play consistent with τ' does not reach a vertex with memory state $m_\uparrow$. Thus, this strategy guarantees a loss less than d. Hence, also $v_M(\mathcal{G}') < d$.

Let τ_{opt} be a positional strategy guaranteeing $v_M(\mathcal{G}')$ for Player 1 in $\mathcal{G}'$. We show that the strategy σ_{opt} induced by $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$ and τ_{opt} is an optimal winning strategy for Player 0 for $\mathcal{G}$. This suffices, since σ_{opt} is finite-state with memory $\mathfrak{M}$ and effectively computable by Theorem 2.20.

By the remarks above, we have $v_R(\sigma_{opt}) = v_M(\mathcal{G}')$. To conclude the proof, we assume towards a contradiction that σ_{opt} is not optimal, i.e., there is a strategy σ for Player 0 for $\mathcal{G}$ such that $v_R(\sigma) < v_R(\sigma_{opt})$. By Lemma 3.20, we can assume without loss of generality that σ uniformly bounds the waiting time for condition j to b_j. Then, the strategy τ' for Player 1 in $\mathcal{G}'$ induced by σ guarantees $v_R(\sigma) < v_R(\sigma_{opt}) = v_M(\mathcal{G}')$, which is the contradiction we were looking for, since τ_{opt} is optimal for $\mathcal{G}'$. □

Corollary 3.22. *The value $v_R(\sigma)$ of an optimal strategy for Player 0 is effectively computable.*

Proof. Construct $\mathcal{G}'$ and compute $v_M(\mathcal{G}')$. If $v_M(\mathcal{G}') = d$, then Player 0 loses $\mathcal{G}$ and $v_R(\sigma) = \infty$ for every strategy σ for Player 0. Otherwise, let σ be the optimal strategy for $\mathcal{G}$ from Theorem 3.7. The values $v_M(\mathcal{G}')$ and $v_R(\sigma)$ coincide. □

Chapter 4

Poset Games

Request-Response Games offer many desirable properties like an intuitive winning condition suitable for real-life applications, finite-state determinacy, and the existence of finite-state time-optimal winning strategies. However, their winning condition is not flexible enough to model many interesting synthesis problems, since it can only specify a single event that answers a response. Consider the intersection example from Chapter 3 and assume there is also a level crossing at that intersection. Then, if a train arrives, all lights have to be changed to red (independently of each other), then the boom barrier has to be lowered, and then the train may cross the street. The barrier can be raised, after the train has left the crossing. This sequence of events cannot be modeled by a Request-Response winning condition.

To obtain a new class of games with a more flexible winning condition while retaining the good characteristics of Request-Response Games, we replace the responses by a finite set of events. Additionally, we allow a partial ordering of the events, as it is often given in planning problems. Then, Player 0 has to satisfy the events in an order that is compatible with the required ordering. Mathematically speaking, the responses form a finite poset and Player 0's goal is to visit vertices of the arena that allow an embedding of the requested poset. The events triggered by an approaching train can be modeled by a poset. Lowering the boom barrier has to be preceded by changing all lights to red, and is followed by an all-clear signal for the train. Finally, the boom may be raised only if the train has left the crossing.

As intended, the Poset winning condition carries over the intuitive notion of waiting times: every time a request is encountered, a clock is started that is not stopped until all events have been satisfied in a compatible order. Compared to Request-Response Games, there is a conceptual difference. In the case of overlapping embeddings there might be several active clocks for a single condition. In contrast, in Request-Response Games there is a single clock for every condition. Requests that are encountered, while another request of that condition is still open, are ignored. Since Request-Response Games are a special case of Poset Games, the framework with multiple clocks is applicable to them as well. This establishes an alternative for defining time-optimal strategies for Request-Response Games.

The value of a play in a Poset Game is the limit of the average accumulated value of all active clocks. So, there is a new kind of trade-off: it still might be worthwhile to keep a request open longer than it has to be, in order to respond to another condition more quickly. But it might also be worthwhile to accept several requests of the same condition while responding to another request. As discussed earlier, this approach penalizes Player 0 for every request that is encountered, not only for the first one. This seems to be reasonable in many applications like the traffic light example from above or an elevator system. In these examples, multiple requests indicate importance of the corresponding response and should be answered first (assuming that the users do not try to cheat the system).

Another example that motivates the analysis of Poset Games is an elevator system, where one can ensure that certain floors are served first (the executive floor, for example). Lastly, aspects of planning can be modeled by a Poset winning condition. In Critical-Path scheduling [30], a planning problem consists of a finite set of tasks and some ordering relation between them. This ordering could state that the roof of a house can only be build if the walls are brought up completely, while the mail box and the windows are independent of each other. To model planning in an infinite game, the arena has designated vertices that allow to begin a certain task and others that signal the completion of a task. The Poset winning condition can be used to specify the ordering constraints of the tasks. Then, Player 0 has to determine a plan for every request. Note that the embeddings might overlap, however. Hence, she wins by constructing one and a half houses, if a second house is requested while she is only half done with the first house. These anomalies have to be ruled out by manipulating the arena, since the Poset winning condition is not expressive enough to prohibit such events.

We begin this chapter with some background from Order Theory and define Poset Games in Section 4.1. Afterwards, we solve Poset Games by reducing them to Büchi Games in Section 4.2. This implies determinacy of Poset Games with finite-state strategies. Afterwards, we define our framework for time-optimal strategies by introducing waiting times in Section 4.3 and state the main theorem of this chapter: if Player 0 wins a Poset Game, then she also has an optimal winning strategy, which is finite-state and effectively computable. We adapt the corrected proof technique for Request-Response Games due to Wallmeier, Horn et. al. [53, 29], which we presented in Chapter 3, to obtain similar results for Poset Games: we show that Player 0 can skip loops, if a request is open for a long time. Iterating this, we show that for every winning strategy for Player 0, there is another winning strategy whose value is not higher and additionally bounds the waiting times for all conditions. This is especially true for an optimal strategy, which therefore can be assumed to bound all waiting times. The improvement of winning strategies is presented in Subsection 4.3.1. As the optimal winning strategy bounds all waiting times, it can be found by a reduction to Mean-Payoff Games presented in Subsection 4.3.2. Since a Request-Response Game can be seen as a Poset Game, there are two frameworks for defining time-optimal winning strategies for Request-Response Games. We close this chapter by discussing the differences between the two approaches.

4.1 Posets and Poset Games

We begin by introducing exactly the amount of Order Theory we need to deal with Poset Games in the remainder of this chapter. These games are then defined formally to close this section.

A *partially ordered set*, or *poset* for short, $\mathcal{P} = (D, \preccurlyeq)$ consists of a non-empty *domain* D and a binary relation $\preccurlyeq$ over D that is

- *reflexive*: $d \preccurlyeq d$ for all $d \in D$,
- *antisymmetric*: $d \preccurlyeq d'$ and $d' \preccurlyeq d$ implies $d = d'$, and
- *transitive*: $d \preccurlyeq d'$ and $d' \preccurlyeq d''$ implies $d \preccurlyeq d''$.

A *labeled poset* $(D, \preccurlyeq, l)$ is a poset with an additional labeling function $l : D \to P$ for a set P of atomic propositions. If D is finite, we define the *transitive reduction* $\preccurlyeq^{red}$ by $d \preccurlyeq^{red} d'$ iff $d \neq d'$ and there is no d'' such that $d'' \neq d$, $d'' \neq d'$, and $d \preccurlyeq d'' \preccurlyeq d'$. The reduction $\preccurlyeq^{red}$ contains all the essential information of $\preccurlyeq$, i.e., the reflexive and transitive closure of $\preccurlyeq^{red}$ is $\preccurlyeq$. The transitive reduction is also the basis of Hasse diagrams, a graphical representation of finite posets. In such a diagram, the elements of the domain are drawn as vertices and a directed edge from d to d' denotes the ordering relation $d \preccurlyeq^{red} d'$. By reducing $\preccurlyeq$, most posets can be clearly represented in a graphic without losing any information.

Example 4.1. $(2^S, \subseteq)$ is a poset for every set S. A Hasse Diagram for $(2^{\{a,b\}}, \subseteq)$ is given in the upper part of Figure 4.1. The solid edges represent the transitive reduction graphically.

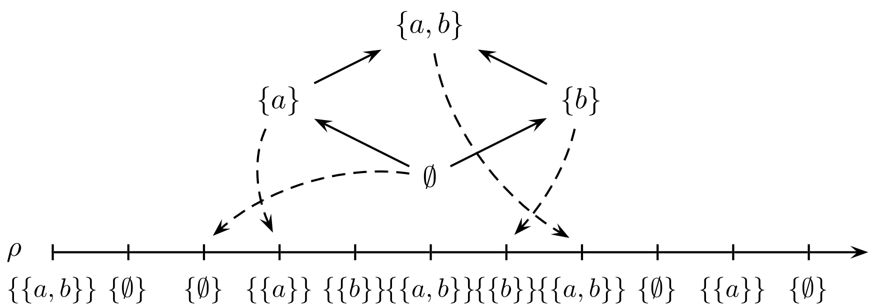

Figure 4.1: A poset $\mathcal{P}$ and an embedding f of $\mathcal{P}$ in ρ

A subset $D' \subseteq D$ of a poset $\mathcal{P} = (D, \preccurlyeq)$ is *upwards-closed*, if $d \in D'$ and $d \preccurlyeq d'$ imply $d' \in D'$. Dually, $D' \subseteq D$ is *downwards-closed*, if $d \in D'$ and $d' \preccurlyeq d$ imply $d' \in D'$. The complement of an upwards-closed subset is downwards-closed, and vice versa. We denote the set of non-empty upwards-closed subsets of D by $\mathrm{Up}(\mathcal{P})$ and the set of non-empty downwards-closed subsets by $\mathrm{Down}(\mathcal{P})$. The number of non-empty downwards-closed (and upwards-closed) subsets of a poset can be bounded from below by $|D|$ and from above by $2^{|D|} - 1$. Notice that we disregard the empty set.

Given a labeled graph (V, E, l_G), a path $\rho = \rho_0\rho_1\rho_2\ldots$ of G, and a finite labeled poset $\mathcal{P} = (D, \preccurlyeq, l_P)$, an *embedding* of $\mathcal{P}$ in ρ is a function $f : D \to \mathbb{N}$ such that $l_P(d) \in l_G(\rho_{f(d)})$ and $d \preccurlyeq d'$ implies $f(d) \leq f(d')$. The *length of an embedding* f is $\max_{d\in D} f(d)$, and it is *minimal*, if its length is minimal in the set of all embeddings, i.e., $\max_{d\in D} f(d) \leq \max_{d\in D} f'(d)$ for all embeddings f'. An embedding of $\mathcal{P}$ in a finite path w is defined analogously.

Example 4.2. Figure 4.1 shows an embedding f of $\mathcal{P} = (2^{\{a,b\}}, \subseteq, l)$ where $l(S) = S$ for every $S \subseteq \{a, b\}$. The labeling of ρ is given by the subsets below the positions. The length of f is 7, but it is not a minimal embedding.

Every subset $D' \subseteq D$ induces a poset by restricting $\preccurlyeq$ to $D' \times D'$. Thus, closed sets and embeddings are also defined for subsets of a poset. Furthermore, every embedding of a downwards-closed subset D' in a finite play x can be completed to an embedding of D in xy, if $D \backslash D'$ can be embedded in y. Hence, a poset can be embedded element by element.

To define Poset Games, we generalize Request-Response Games by replacing the responses by labeled posets, and require that a request is responded by an embedding of that poset. A (initialized) *Poset Game* $(G, s_0, (q_j, \mathcal{P}_j)_{j=1,\ldots,k})$ consists of a labeled arena G with labeling function l_G, an initial vertex s_0, and a finite collection of *(Request-Poset) conditions* $(q_j, \mathcal{P}_j)$, where the *request* $q_j \in P$ is a proposition and $\mathcal{P}_j = (D_j, \preccurlyeq_j, l_j)$ is a finite, labeled poset. We assume tacitly that the domains of the posets are pairwise disjoint. Player 0's goal is to answer every request q_j by an embedding of $\mathcal{P}_j$. Thus, we define $\rho = \rho_0\rho_1\rho_2\ldots \in \mathrm{Win}$ iff

$$\forall j\, \forall n\, (q_j \in l_G(\rho_n) \to \rho_n\rho_{n+1}\rho_{n+2}\ldots \text{ allows an embedding of } \mathcal{P}_j)\,.$$

Again, a *request of condition* j is a vertex s such that $q_j \in l_G(s)$. A *response* of that request is a finite play w starting in s that allows an embedding of $\mathcal{P}_j$. If w does not allow an embedding, then the request is still *open* after w. However, unlike in Request-Response Games, the notion of an open request is no longer binary, but it can be refined: we say that $D \subseteq D_j$ is *open* after the finite play $\rho_0 \ldots \rho_n$, if there was a request at position $k \leq n$ such that the elements in $D_j \backslash D$ could be embedded in $\rho_k \ldots \rho_n$, but no superset of $D_j \backslash D$ could be embedded in this suffix. This means Player 0 was able to embed the elements of $D_j \backslash D$ (which form a downwards-closed subset by the requirements on an embedding) and the elements of D (which form an upwards-closed subset) have to be embedded yet.

Example 4.3. Let $D_j = \{d_1, d_2, d_3\}$, $\preccurlyeq_j$ be specified by $d_1 \preccurlyeq_j^{red} d_2 \preccurlyeq_j^{red} d_3$, and $l_j(d) = d$ for all $d \in D_j$. Finally, let $(q_j, (D_j, \preccurlyeq_j, l_j))$ be a condition and ρ as in Figure 4.2.

The positions of ρ are denoted above and the sets below are the labels of each position. The vertices ρ_0, ρ_3, ρ_6, and ρ_7 are requests, D_j is open after the finite play $\rho_0\rho_1$, and there are three open requests after $\rho_0 \dots \rho_6$: The set D_j from the request at ρ_6, $\{d_2, d_3\}$ from the request at ρ_3, and $\{d_3\}$ from the request at ρ_0. All requests are responded completely with ρ_9.

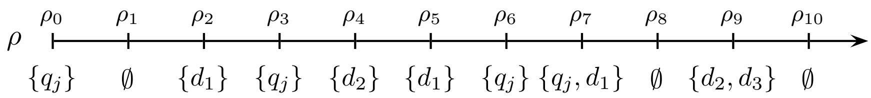

Figure 4.2: A play with (open) requests

4.2 Solving Poset Games

In this section, we solve Poset Games by a reduction to Büchi Games. The idea is to use memory to keep track of the requests that have not yet been responded completely. However, it is not sufficient to store a subset of D_j only, which represents open elements that still have to be embedded, and require that no element stays in that set indefinitely.

Example 4.4. Consider a game with a single condition composed of a request q and a poset $(\{d_1, d_2\} \preccurlyeq)$ with $d_1 \preccurlyeq d_2$ (for the sake of readability, we omit the labeling functions), and the play ρ in Figure 4.3. After every finite prefix of ρ, d_2 has to be embedded once more, since every time it is embedded and the latest request is responded, it is requested again, but cannot be embedded, since d_1 blocks it.

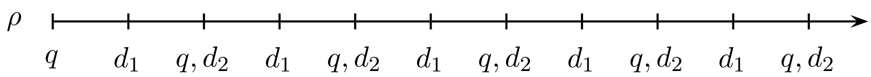

Figure 4.3: A play ρ with overlapping embeddings

This example demonstrates the need for a finer memory structure to store open requests. For every request of condition j, we keep track of those elements of D_j that are already embedded and those that are still open. Since there is a set for every request, we can determine, whether every request is responded eventually. By adding a clock to every such set, which measures the length of the embedding, we lay the groundwork for defining the waiting times in Section 4.3.

Formally, we define the *set of open requests of condition j after the finite play w* inductively by $\mathrm{Open}_j(\varepsilon) = \emptyset$ and

$$\mathrm{Open}_j(ws) = \{(\mathrm{Emb}_j(D,s), t+1) \mid (D,t) \in \mathrm{Open}_j(w) \cup \{(\mathrm{New}_j(s), 0)\}\} \setminus \{\emptyset\} \times \mathbb{N}$$

where

$$\mathrm{New}_j(s) = \begin{cases} D_j & \text{if } q_j \in l_G(s) \\ \emptyset & \text{otherwise} \end{cases}$$

and

$$\mathrm{Emb}_j(D,s) = \{d \in D \mid \exists d' \in D : d' \preccurlyeq_j d \text{ and } l_j(d') \notin l_G(s)\}.$$

New_j signals a new request and adds the domain D_j to the set of open requests. To compute the embedding monotonically, Emb_j determines those elements of $D \in \mathrm{Up}(\mathcal{P}_j)$ that cannot be embedded in s since their or some smaller element's labeling is not compatible with the labeling of s. Finally, Open_j takes the open requests of w and embeds as many elements as possible into s, thereby determining the open requests of ws: If $(D, t+1) \in \mathrm{Open}_j(\rho_0 \dots \rho_n)$, then there was a request of condition j at position $n-t$, and the elements of $D_j \setminus D$ can be embedded into $\rho_{n-t} \dots \rho_n$, and Player 0 has to embed all elements of D to respond to the request. For every t, there is at most one $D \in \mathrm{Up}(\mathcal{P}_j)$ such that $(D,t) \in \mathrm{Open}_j(w)$. Also, $\mathrm{Open}_j(w)$ contains only upwards-closed subsets of D_j, which can be shown by an easy induction. The *number of open requests D of condition j after w* is

$$s_{j,D}(w) = |\{t \mid (D,t) \in \mathrm{Open}_j(w)\}|.$$

Example 4.5. Consider the play ρ from Example 4.4 once more: The open requests $\mathrm{Open}_j(w)$ toggle between $\{(\{d_1, d_2\}, 1)\}$ and $\{(\{d_2\}, 2)\}$

We say that $D \in \mathrm{Up}(\mathcal{P}_j)$ is *open indefinitely* in $\rho = \rho_0\rho_1\rho_2\dots$, if there exists n such that $(D,t) \in \mathrm{Open}_j(\rho_0 \dots \rho_{n+t})$ for all t. In the next lemma we show that this characterizes the losing plays for Player 0.

Lemma 4.6. *Let $\rho = \rho_0\rho_1\rho_2\dots$ be a play and $j \in [k]$.*

(i) If Player 0 wins ρ, then $(\mathrm{Open}_j(\rho_0 \dots \rho_n))_{n\in\mathbb{N}}$ induces a minimal embedding f_n of $\mathcal{P}_j$ in $\rho_n\rho_{n+1}\rho_{n+2}\dots$ for every n such that $q_j \in l_G(\rho_n)$.

(ii) ρ is won by Player 0 iff there is no $D \in \mathrm{Up}(\mathcal{P}_j)$ that is D open indefinitely.

Proof. We begin by defining the sequence $(D_{n,t})_{t\in\mathbb{N}}$ for every request of condition j at position n by $D_{n,0} = \mathrm{Emb}_j(D_j, \rho_n)$ and $D_{n,t+1} = \mathrm{Emb}_j(D_{n,t}, \rho_{n+t+1})$. We have $D_{n,t} \supseteq D_{n,t+1}$ and $(D_{n,t}, t+1) \in \mathrm{Open}_j(\rho_0 \dots \rho_{n+t})$, if $D_{n,t} \neq \emptyset$.

(i) Let $q_j \in l_G(\rho_n)$ and let f be a minimal embedding of $\mathcal{P}_j$ in $\rho_n\rho_{n+1}\rho_{n+2}\ldots$, whose existence is guaranteed, since ρ is won by Player 0. We define the embedding f_n by $f_n(d) = \min\{t \mid d \notin D_{n,t}\}$. We show that f_n is well-defined and minimal by proving $d \notin D_{n,f(d)}$ by Noetherian induction (on $\preccurlyeq_j^{red}$) over d. Let $d \in D_j$. Towards a contradiction, assume $d \in D_{n,f(d)}$.

If $f(d) = 0$, then $d \in D_{n,0} = \mathrm{Emb}_j(D_j, \rho_n)$, i.e., there exists a $d' \in D_j$ such that $d' \preccurlyeq_j d$ and $l_j(d') \notin l_G(\rho_n)$. Since $d' \preccurlyeq_j d$ implies $f(d') = 0$, and we conclude $l_j(d') \in l_G(\rho_{n+f(d')})$, which yields the desired contradiction.

If $f(d) = t+1$, then $d \in D_{n,t+1} = \mathrm{Emb}_j(D_{n,t}, \rho_{n+t+1})$, i.e., there exists a $d' \in D_{n,t}$ such that $d' \preccurlyeq_j d$ and $l_j(d') \notin l_G(\rho_{n+t+1})$. Since f is an embedding, $l_j(d) \in l_G(\rho_{n+f(d)})$, which rules out $d' = d$. Thus, $d' \preccurlyeq_j^{red} d$, which allows us to apply the induction hypothesis to obtain $d' \notin D_{n,f(d')}$. If $f(d') < t+1$, then $d' \notin D_{n,t}$, since $D_{n,f(d')} \supseteq D_{n,t}$. On the other hand, $f(d') = t+1$ implies $l_j(d') \in l_G(\rho_{n+t+1})$. So, both cases yield the desired contradiction.

(ii) We have

$$\mathrm{Open}_j(\rho_0 \ldots \rho_n) = \{(D_{n',t}, t+1) \mid n' + t = n,\ q_j \in l_G(\rho_{n'}),\ \text{and } D_{n',t} \neq \emptyset\},$$

which can be verified by an easy induction. In (i) we have seen that $(D_{n,t})_{n\in\mathbb{N}}$ converges to the empty set for every n, if Player 0 wins ρ. Thus, there cannot be a $D \in \mathrm{Up}(\mathcal{P}_j)$ that is open indefinitely. On the other hand, if there is no $D \in \mathrm{Up}(\mathcal{P}_j)$ that is open indefinitely, then $(D_{n,t})_{n\in\mathbb{N}}$ converges to the empty set for every n. We define an embedding f_n for every n such that $q \in l_G(\rho_n)$, analogously to the one defined in (i) by $f_n(d) = \min\{t \mid d \notin D_{n,t}\}$. It remains to show that f_n is an embedding: The element d leaves $(D_{n,t})_{n\in\mathbb{N}}$ at $f_n(d)$ iff the labeling requirement is fulfilled, by definition of Emb_j. Also,

$$\min\{t \mid d' \notin D_{n,t}\} = f_n(d') > f_n(d) = \min\{t \mid d \notin D_{n,t}\}$$

for $d' \preccurlyeq_j d$ contradicts the definition of Emb_j. □

We now prove that Poset Games are reducible to Büchi Games. This implies determinacy with finite-state strategies. Additionally, this reduction gives an upper bound on the value of an optimal winning strategy for Player 0, a corollary stated in Section 4.3 after having introduced waiting times and values of strategies.

Theorem 4.7. *Poset Games are reducible to Büchi Games.*

The first idea is to use the memory to store the sets $open_j$ (without the clocks to obtain a finite memory structure) and employ a cyclic counter to ensure that no D is open indefinitely. While this is correct due to Lemma 4.6, we can do better than a double exponential blow-up. We have seen in Example 4.4 that just keeping track of the elements of the domains that still have to be embedded does not generate enough information to decide the winner of a play. An element could be in that set since it is either requested and embedded infinitely often (as in the example), or it cannot be embedded at all. The extra information needed can be generated by marking every position the element could be mapped to. In the example, there are infinitely many positions of ρ that allow an embedding of d_2. Thus, we need two different subsets of D_j: The set O_j contains the elements that still have to be embedded, i.e., the union of the sets in Open_j. The set M_j contains exactly those elements that could be embedded in the last vertex of the finite play. Then, a play is good for Player 0 if every element that is indefinitely in O_j is infinitely often contained in M_j. This can be checked using a cyclic counter.

Proof. We begin the definition of the memory structure with the counter that cycles through the elements of the domains. By enumerating D_j consecutively in an order compatible with $\preccurlyeq_j$ we are able to give a better bound on the value of the induced strategy. Therefore, let $e_j : [|D_j|] \to D_j$ be an enumeration of D_j such that $d \preccurlyeq d'$ implies $e_j^{-1}(d) \leq e_j^{-1}(d')$, let $l = \sum_{j=1}^{k} |D_j|$, and define $e : [l] \to \bigcup_{j=1}^{k} \{j\} \times D_j$ by

$$e\left(\left(\sum_{j=1}^{j'-1} |D_j|\right) + h\right) = (j', e_{j'}(h)) \quad \text{for } h \in [|D_{j'}|].$$

e is an enumeration with the desired properties. We assume without loss of generality $l > 1$, since the counter does not cycle if $l = 1$.

We define the memory structure $\mathfrak{M} = (M, \mathrm{init}, \mathrm{update})$ with memory

$$M = \prod_{j=1}^{k} (\mathrm{Up}(\mathcal{P}_j) \times 2^{D_j}) \times [l] \times \{0, 1\},$$

initialization function

$$\mathrm{init}(s) = (\mathrm{Emb}_1(\mathrm{New}_1(s)), \emptyset, \ldots, \mathrm{Emb}_k(\mathrm{New}_k(s)), \emptyset, 1, 0),$$

and update function

$$\mathrm{update}((O_1, M_1, \ldots, O_k, M_k, m, f), s) = (O'_1, M'_1, \ldots, O'_k, M'_k, m', f')$$

where

- $O'_j = \begin{cases} \mathrm{Emb}_j(D_j, s) & \text{if } q_j \in l_G(s) \\ \mathrm{Emb}_j(O_j, s) & \text{if } q_j \notin l_G(s) \end{cases},$
- $M'_j = \{d \in O'_j \mid l_j(d) \in l_G(s)\},$

- $m' = \begin{cases} (m \mod l) + 1 & \text{if } e(m) = (j, d) \text{ and } d \notin O'_j \text{ or } d \in M'_j \\ m & \text{if } e(m) = (j, d) \text{ and } d \in O'_j \text{ and } d \notin M'_j \end{cases}$, and
- $f' = \begin{cases} 1 & \text{if } m \neq m' \\ 0 & \text{otherwise} \end{cases}$.

It is easy to verify that the update function is well-defined, i.e., O'_j is upwards-closed if O_j is upwards-closed. Finally, let

$$F = V \times \prod_{j=1}^{k} (\mathrm{Up}(\mathcal{P}_j) \times 2^{D_j}) \times [l] \times \{1\}$$

and $\mathcal{G}' = (G \times \mathfrak{M}, F)$ be the Büchi Game in the expanded arena.

It remains to show $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$. Let $\rho = \rho_0\rho_1\rho_2\dots$ be a play of $\mathcal{G}$ and $\rho' = \rho'_0\rho'_1\rho'_2\dots$ the unique expanded play in $\mathcal{G}'$ where $\rho'_n = (\rho_n, (O_1^n, M_1^n, \dots, O_k^n, M_k^n, m^n, f^n))$.

Let ρ be winning for Player 0 and assume towards a contradiction that ρ' is winning for Player 1. Then, the counter stops at some position n' with some value c and does not change anymore. This means $d \in O_j^n$ and $d \notin M_j^n$ for all $n \geq n'$, where $e(c) = (j, d)$. If condition j is requested infinitely often in ρ, then $\mathcal{P}_j$ is embedded infinitely often in ρ. Thus, there are infinitely many n such that $d \in M_j^n$, which yields the desired contradiction. On the other hand, if there is a final request at position n, then there is also an embedding f of $\mathcal{P}_j$ in $\rho_n\rho_{n+1}\rho_{n+2}\dots$. It holds $O_j^{n+t} = \{d \in D_j \mid f(d) > t\}$. Thus, the O_j-component is empty from some position onwards, which again is a contradiction.

Now, let ρ' be winning for Player 0. For every n such that $q_j \in l_G(\rho_n)$, we have to construct an embedding f_n of $\mathcal{P}_j$ in $\rho_n\rho_{n+1}\rho_{n+2}\dots$. Since ρ' is won by Player 0, there are infinitely many positions n' such that $d \notin O_j^{n'}$ or $d \in M_j^{n'}$.

Let $q_j \in l_G(\rho_n)$. We define the sequence $(D_{n,t})_{t\in\mathbb{N}}$ by $D_{n,0} = \mathrm{Emb}_j(D_j, \rho_n)$ and $D_{n,t+1} = \mathrm{Emb}_j(D_{n,t}, \rho_{n+t+1})$. We have $D_{n,t} \subseteq O_j^{n+t}$ for all t, which can be verified by an easy induction. Also, $D_{n,t} \supseteq D_{n,t+1}$ for all t. Now, if $D_{n,t} = \emptyset$ for some t, then the sequence induces f_n by $f_n(d) = \min\{t \mid d \notin D_{n,t}\}$ as we have shown in the proof of Lemma 4.6 (ii). Hence, it remains to be shown that for every $d \in D_j$ there is a t such that $d \notin D_{n,t}$.

Towards a contradiction, assume there exists t such that $\emptyset \neq D_{n,t} = D_{n,t'}$ for all $t' \geq t$, and let d be minimal in $D_{n,t}$, i.e., there is no $d' \neq d$ such that $d' \preccurlyeq_j d$ and $d' \in D_{n,t}$. Thus, $d \in O_j^{n+t'}$ for all $t' \geq t$. Since ρ' is winning for Player 0, there is some $t' > t$ such that $d \in M_j^{n+t'}$, which implies that d can be embedded in $\rho_{n+t'}$, i.e., we have $d \notin \mathrm{Emb}_j(D_{n,t'-1}, \rho_{n+t'}) = D_{n,t'}$, which yields the desired contradiction. □

Note that by the choice of e, it takes at most $l + |D_j|$ visits to a vertex of F after a request to complete the embedding f_n.

$\mathfrak{M}$ is essentially a deterministic automaton, which can be turned into a Büchi Automaton with set F of accepting states defined similarly to the one in the proof above.

Corollary 4.8. *Poset Games are regular.*

In an implementation the size of the memory structure can be reduced drastically since every state with $M_j \not\subseteq O_j$ is not reachable. However, for our purposes this optimization is not necessary. The size of $\mathfrak{M}$ can be bounded by

$$\begin{aligned}
|M| &= \left(\prod_{j=1}^{k} |\mathrm{Up}(\mathcal{P}_j)| \cdot 2^{|D_j|}\right) \cdot \left(\sum_{j=1}^{k} |D_j|\right) \cdot 2 \\
&\leq \left(\prod_{j=1}^{k} 2^{2\cdot|D_j|}\right) \cdot l \cdot 2 \\
&= \left(2^{2\cdot\sum_{j=1}^{k}|D_j|}\right) \cdot l \cdot 2 \\
&= l \cdot 2^{2\cdot l+1}
\end{aligned}$$

We finish this section by proving that the reduction presented here is asymptotically optimal.

Lemma 4.9. *There is a family of initialized Poset Games $(\mathcal{G}_n)_{n\in\mathbb{N}}$ such that*

(i) the size of the arena of $\mathcal{G}_n$ is linear in n,

(ii) the sum of the cardinalities of the posets' domains is linear in n,

(iii) Player 0 wins $\mathcal{G}_n$, but

(iv) she has no finite-state winning strategy of size less than $n \cdot 2^n$.

Proof. Every Request-Response Game with k Request-Response conditions is equivalent to a Poset Game with k Poset conditions whose domains are singletons. So, we can use the games from Lemma 3.2 and translate them into Poset Games $\mathcal{G}_n$. The first three statements are clear. For the last claim, assume Player 0 has a finite-state strategy of size smaller than $n \cdot 2^n$. This is also a small finite-state winning strategy for the original Request-Response Game, which yields a contradiction. □

4.3 Time-optimal Strategies for Poset Games

In this section we define waiting times for Poset Games and values for strategies, discuss some simple properties, and state the main theorem about Poset Games.

Unlike waiting times for Request-Response Games, which ignore a request while an earlier request of the same condition is already open, waiting times for Poset Games are defined for every request. For Request-Response Games there is a single clock for every condition. It is started if a request occurs and stops as soon as it is responded, ignoring all intermediate requests. For Poset Games, a clock is started for every request. Hence, there are possibly several active clocks for a single condition at the same time. We introduced these clocks already when we defined the set of open requests Open_j. To aggregate the clock values to the value of a play, we take the limit of the accumulated sum of the different clocks for all open requests. Again, we employ a family of strictly increasing *penalty functions* $(f_j : \mathbb{N} \to \mathbb{N})_{j=1\dots k}$ to be able to prioritize certain conditions. We define for $D \in \mathrm{Up}(\mathcal{P}_j)$

- the *totalized waiting time for D after w*: $t_{j,D}(w) = \sum_{(D,t)\in\mathrm{Open}_j(w)} t$,
- the *totalized waiting time for condition j after w*: $t_j(w) = \sum_{D\in\mathrm{Up}(\mathcal{P}_j)} t_{j,D}(w)$,
- the *penalty for condition j after w*: $p_j(w) = f_j(t_j(w))$,
- the *penalty after w*: $p(w) = \sum_{j=1}^{k} p_j(w)$,
- the *value of a play ρ*: $v_P(\rho) = \limsup_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} p(\rho_0 \dots \rho_i)$, and
- the *value of a strategy σ*: $v_P(\sigma) = \sup_{\tau\in\Gamma_1} v_P(\rho(s_0, \sigma, \tau))$.

A strategy σ for Player 0 is *optimal* if $v_P(\sigma) \leq v_P(\sigma')$ for all strategies σ'.

The following Lemma is the analogon of Lemma 3.5 for Poset Games and its proof goes along the same lines: if Player 1 wins a play, then the waiting times increase without a bound and its value diverges.

Lemma 4.10. *Let ρ be a play and σ a strategy for Player 0.*

(i) If $v_P(\rho) < \infty$, then Player 0 wins ρ.

(ii) If $v_P(\sigma) < \infty$, then σ is a winning strategy for Player 0.

Note that the other directions of the statements are false. There is a play of infinite value that is won by Player 0.

There are two different kinds of bounds on the waiting time. Firstly, the clocks can be bounded which implies that the length of every embedding is bounded. Secondly, the sum of the clocks can be bounded which indirectly bounds the lengths of the embeddings. Both approaches are intimately tied and we use their interplay throughout this chapter. Let σ be a strategy for Player 0 and $D \in \mathrm{Up}(\mathcal{P}_j)$ for some condition j.

- *σ uniformly bounds the waiting time for D to B*, if $t \leq B$ for all $(D, t) \in \mathrm{Open}_j(w)$ and for all finite plays w consistent with σ.
- *σ uniformly bounds the totalized waiting time for D to B*, if $t_{j,D}(w) \leq B$ for all finite plays w consistent with σ.

The following statements are easily derived from the definitions above.

Remark 4.11. *Let σ be a strategy for Player 0.*

(i) If σ uniformly bounds the waiting time for all $D \in \mathrm{Up}(\mathcal{P}_j)$ to B, then every request of condition j is responded by an embedding of $\mathcal{P}_j$ whose length is at most B.

(ii) If σ uniformly bounds the totalized waiting time for all $D \in \mathrm{Up}(\mathcal{P}_j)$ to B, then every request of condition j is responded by an embedding of $\mathcal{P}_j$ whose length is at most B.

(iii) If σ uniformly bounds the waiting time for D to B, then σ also uniformly bounds the totalized waiting time for D to $\frac{B\cdot(B+1)}{2}$.

(iv) If σ uniformly bounds the totalized waiting time for D to B, then σ also uniformly bounds the waiting time for D to B.

Now, we can pick up the reduction of Poset Games to Büchi Games and show that if Player 0 wins $\mathcal{G}$, then the value of an optimal strategy can be bounded in terms of the size of the arena and in the size of the winning condition.

Corollary 4.12. *Let $l = \sum_{j=1}^{k} |D_j|$. If Player 0 wins $\mathcal{G}$, then*

$$v_P(\sigma) \leq \sum_{j=1}^{k} f_j \left(|\mathrm{Up}(\mathcal{P}_j)| \frac{|G|(l+|D_j|)(|G|(l+|D_j|)+1)}{2} \right) =: b_P(\mathcal{G})$$

for an optimal strategy σ.

Proof. We go along the lines of the proof of Corollary 3.6. Let σ' be the positional winning strategy for $\mathcal{G}'$ from Theorem 4.7 and σ be the induced finite-state strategy for $\mathcal{G}$. There is no infix of length $|G|$ of a play ρ' played according to σ' that does not visit F at least once. If there is such an infix, then there is a loop in that infix in which no vertex of F is visited. Moving through that loop indefinitely is consistent with σ'. Thus, it is not a winning strategy for Player 0, contrary to our assumptions. Therefore, the counter m changes its value after at most $|G|$ steps. In the proof of Theorem 4.7 the counter is constructed in a way that it takes at most $l + |D_j|$ visits to a state in F to complete an embedding after a request in the projected play. Hence, the length of every embedding of $\mathcal{P}_j$ in a play consistent with σ is bounded by $|G| \cdot (l + |D_j|)$, which gives

$$t_{j,D}(w) \leq \frac{|G| \cdot (l+|D_j|) \cdot (|G| \cdot (l+|D_j|)+1)}{2}$$

for every finite play w consistent with σ, by Remark 4.11 (iii). We obtain

$$\frac{1}{n}\sum_{i=0}^{n-1} p(\rho_0 \dots \rho_i) = \frac{1}{n}\sum_{i=1}^{n-1}\sum_{j=1}^{k} f_j \left(\sum_{D \in \mathrm{Up}(\mathcal{P}_j)} t_{j,D}(\rho_0 \dots \rho_i) \right)$$

$$\leq \frac{1}{n}\sum_{i=1}^{n-1}\sum_{j=1}^{k} f_j \left(\sum_{D\in \mathrm{Up}(\mathcal{P}_j)} \frac{|G|(l+|D_j|)(|G|(l+|D_j|)+1)}{2} \right)$$

$$= \sum_{j=1}^{k} f_j \left(|\mathrm{Up}(\mathcal{P}_j)| \frac{|G|(l+|D_j|)(|G|(l+|D_j|)+1)}{2} \right) = b_P(\mathcal{G})$$

for every play $\rho = \rho_0\rho_1\rho_2\ldots$ that is played according to σ. Hence, $v_P(\rho) \leq b_P(\mathcal{G})$ for every play ρ consistent with σ. □

We now state the main theorem of this chapter and spend the next two subsections proving it. In Subsection 4.3.1, we show that an optimal winning strategy uniformly bounds the totalized waiting times for all $D \in \mathrm{Up}(\mathcal{P}_j)$. This allows us to reduce the problem of searching an optimal strategy for Poset Games to the same problem for Mean-Payoff Games, which is presented in Subsection 4.3.2. The proof idea is similar to the corresponding proof for Request-Response Games; however, we have to refine the first step to overcome the possible overlapping of embeddings.

Theorem 4.13. *If Player 0 wins a Poset Game $\mathcal{G}$, then she also has a finite-state optimal winning strategy which is effectively computable.*

4.3.1 Strategy Improvement for Poset Games

In this Subsection, we do the first step of proving Theorem 4.13. Therefore, we adapt the technique described in Subsection 3.2.1 to Poset Games. In a play of a Request-Response Game, Player 0 skips a loop if the waiting time for condition j exceeds a certain bound and the condition is not responded in that loop, i.e., the loop is between a request and a response. Additionally, she has to take care that she does not miss responses of the other conditions. Therefore, she only skips a loop if the waiting times for all other conditions are higher at the end of the loop than they were at the beginning. In a Poset Game the situation is more complicated. Player 0 cannot just skip an arbitrary loop between a request and the end of the corresponding embedding since the positions to which an element is embedded must not be deleted. So, she can only skip loops in between those positions. This is illustrated in Figure 4.4. Player 0 can skip loops in the intervals I between the positions where d_1 and d_2 hold and the positions where d_2 and d_3 hold, respectively.

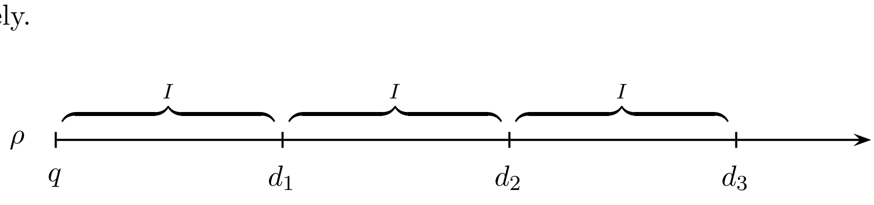

Figure 4.4: The intervals I where Player 0 can improve her strategy

So, we define a strategy improvement operator for each such interval. Again, the waiting times for the other conditions and all of their open requests have to be respected. Thus, we require that the totalized waiting time for every $D' \in \mathrm{Up}(\mathcal{P}_{j'})$ for every j' is higher at the end of the loop than it was at the beginning. However, there is another difference in the waiting time framework. In a Poset Game there might be several active clocks at a given time. Thus, the totalized waiting time can increase by more than one unit in a single step. The correctness of the strategy improvement operator I_j for Request-Response Games was based on Remark 3.3, which states that $t_j(x) \leq t_j(y)$ implies $t_j(xs) \leq t_j(ys)$. An easy induction then proves that skipping loops does only decrease the waiting times. However, this does not hold for the totalized waiting time in a Poset Game: consider finite plays x, y such that $\mathrm{Open}_j(x) = \{(D,1),(D,2)\}$ and $\mathrm{Open}_j(y) = \{(D,3)\}$. Then, $t_{j,D}(x) = 3 = t_{j,D}(y)$. Now, assume that no element of D can be embedded in s. Then, $t_{j,D}(xs) = 4$, but $t_{j,D}(ys) = 3$. So, we have to strengthen the condition that determines the loops to skip. The problem in the example is caused by the fact that there are two active clocks for D at x but only one at y. So, the growth of the totalized waiting time of two clocks outgrows the totalized waiting time of a single clock. It turns out that this is the only reason for a violation of the desired inequality.

Lemma 4.14. *Let $x, y \in V^*$ and $s \in V$ such that $t_{j,D}(x) \leq t_{j,D}(y)$ and $s_{j,D}(x) \leq s_{j,D}(y)$ for all $j \in [k]$ and all $D \in \mathrm{Up}(\mathcal{P}_j)$. Then, $t_{j,D}(xs) \leq t_{j,D}(ys)$ and $s_{j,D}(xs) \leq s_{j,D}(ys)$ for all $j \in [k]$ and all $D \in \mathrm{Up}(\mathcal{P}_j)$.*

Proof. We have

$$
\begin{aligned}
& t_{j,D}(xs) \\
=& \sum_{(D,t)\in \mathrm{Open}_j(xs)} t \\
=& |\{\mathrm{Emb}_j(\mathrm{New}_j(s), s) \cap \{D\}\}| + \sum_{\substack{(D',t')\in \mathrm{Open}_j(x):\\ \mathrm{Emb}(D',s)=D}} (t'+1) \\
=& |\{\mathrm{Emb}_j(\mathrm{New}_j(s), s) \cap \{D\}\}| + \sum_{\substack{D'\in \mathrm{Up}(\mathcal{P}_j):\\ \mathrm{Emb}(D',s)=D}} (t_{j,D'}(x) + s_{j,D'}(x)) \\
\leq& |\{\mathrm{Emb}_j(\mathrm{New}_j(s), s) \cap \{D\}\}| + \sum_{\substack{D'\in \mathrm{Up}(\mathcal{P}_j):\\ \mathrm{Emb}(D',s)=D}} (t_{j,D'}(y) + s_{j,D'}(y)) \\
=& |\{\mathrm{Emb}_j(\mathrm{New}_j(s), s) \cap \{D\}\}| + \sum_{\substack{(D',t')\in \mathrm{Open}_j(y):\\ \mathrm{Emb}(D',s)=D}} (t'+1) \\
=& \sum_{(D,t)\in \mathrm{Open}_j(ys)} t \\
=& t_{j,D}(ys)
\end{aligned}
$$

and

$$
\begin{aligned}
&s_{j,D}(xs)\\
=&|\{t \mid (D,t) \in \mathrm{Open}_j(xs)\}|\\
=&|\{\mathrm{Emb}_j(\mathrm{New}_j(s),s) \cap \{D\}\}| + \sum_{\substack{(D',t')\in\mathrm{Open}_j(x):\\ \mathrm{Emb}(D',s)=D}} |\{t' \mid (D',t') \in \mathrm{Open}_j(x)\}|\\
=&|\{\mathrm{Emb}_j(\mathrm{New}_j(s),s) \cap \{D\}\}| + \sum_{\substack{(D',t')\in\mathrm{Up}(\mathcal{P}_j):\\ \mathrm{Emb}(D',s)=D}} s_{j,D'}(x)\\
\leq&|\{\mathrm{Emb}_j(\mathrm{New}_j(s),s) \cap \{D\}\}| + \sum_{\substack{(D',t')\in\mathrm{Up}(\mathcal{P}_j):\\ \mathrm{Emb}(D',s)=D}} s_{j,D'}(y)\\
=&|\{\mathrm{Emb}_j(\mathrm{New}_j(s),s) \cap \{D\}\}| + \sum_{\substack{(D',t')\in\mathrm{Open}_j(y):\\ \mathrm{Emb}(D',s)=D}} |\{t' \mid (D',t') \in \mathrm{Open}_j(y)\}|\\
=&|\{t \mid (D,t) \in \mathrm{Open}_j(ys)\}|\\
=&s_{j,D}(ys). \qquad \square
\end{aligned}
$$

Note that an open request is always an upwards-closed subset of the domain. Hence, we define the *strategy improvement operator* $I_{j,D}$ for every subset $D \in \mathrm{Up}(\mathcal{P}_j)$. Given a winning strategy σ so that $v_P(\sigma) \leq b_P(\mathcal{G})$, the strategy $I_{j,D}(\sigma)$ is defined using a memory structure $\mathfrak{M} = (M, \mathrm{init}, \mathrm{update})$. The set of memory states M is a subset of the finite plays consistent with σ and defined implicitly. The initialization function is given by $\mathrm{init}(s_0) = s_0$. We define the update function such that the following invariant holds: the last vertices of w and $\mathrm{update}^*(w)$ are equal for every play w consistent with $I_{j,D}(\sigma)$.

Player 0 only skips loops if the totalized waiting time for D is guaranteed to be higher than the value of the strategy. Then, the value of the play does not increase from taking the shortcut. Thus, if $t_{j,D}(ws) \leq f_j^{-1}(b_P(\mathcal{G}))$, let $\mathrm{update}(w, s) = ws$. Hence, if the totalized waiting time is small, then she copies the original play according to σ. Otherwise, if $t_{j,D}(ws) > f_j^{-1}(b_P(\mathcal{G}))$ consider the unraveling $\mathfrak{T}^{\sigma}_{G,s_0}\rceil_{ws}$ restricted to those paths wsx such that $\mathrm{Open}_j(wsx') \cap (\{D\} \times \mathbb{N}) \neq \emptyset$ for all $x' \sqsubseteq x$. This tree is finite, since σ is a winning strategy. The set of finite plays zs ending in s in the restricted unraveling such that $t_{j',D'}(zs) \geq t_{j',D'}(ws)$ and $s_{j',D'}(zs) \geq s_{j',D'}(ws)$ for all $j' \in [k]$ and all $D' \in \mathrm{Up}(\mathcal{P}_{j'})$ is non-empty as it contains ws. Let zs be a play of maximal length in that set. Then, $\mathrm{update}(ws) = zs$. So, if the totalized waiting time for D is sufficiently high, then Player 0 looks ahead for a loop such that the totalized waiting times for all subsets $D' \in \mathrm{Up}(\mathcal{P}_{j'})$ are higher at the end of the loop than they were at the beginning. Then, she jumps ahead and continues to play as if she had finished the loop already. The condition on the subsets D' ensures that she does not miss a vertex that she has to visit in order to embed an element of the posets.

It is clear that the invariant is satisfied in both cases of the definition: the last vertices of $\text{update}^*(w)$ and w are equal. Finally, let $\text{next}(ws, s) = \sigma(ws)$. This concludes the definition of the improved strategy $I_{j,D}(\sigma)$.

The following Lemma is an easy adaptation of the corresponding result for the strategy improvement operator for Request-Response Games. The proof is the same as the one for Lemma 3.8.

Lemma 4.15. $\text{update}^*(w)$ *is consistent with* σ *for all* w *consistent with* $I_{j,D}(\sigma)$.

For the next result, the analogon of Lemma 3.9, we apply Lemma 4.14.

Lemma 4.16. *Let* σ *be a winning strategy for Player* 0 *in* $\mathcal{G}$, $j \in [k]$, *and* $D \in \text{Up}(\mathcal{P}_j)$. *Then,* $t_{j',D'}(w) \leq t_{j',D'}(\text{update}^*(w))$ *for all* w *consistent with* $I_{j,D}(\sigma)$, *for all* $j' \in [k]$ *and all* $D' \in \text{Up}(\mathcal{P}_{j'})$.

Proof. By induction over w, we prove the stronger claim $t_{j',D'}(w) \leq t_{j',D'}(\text{update}^*(w))$ and $s_{j',D'}(w) \leq s_{j',D'}(\text{update}^*(w))$ for all w consistent with $I_{j,D}(\sigma)$. The induction base is clear as every play starts in s_0 and we have $s_0 = \text{init}(s_0) = \text{update}^*(s_0)$. By the induction hypothesis, we can assume

$$t_{j',D'}(w) \leq t_{j',D'}(\text{update}^*(w)) \text{ and } s_{j',D'}(w) \leq s_{j',D'}(\text{update}^*(w))$$

for all $j' \in [k]$ and all $D' \in \text{Up}(\mathcal{P}_{j'})$. Furthermore, Lemma 4.14 gives

$$t_{j',D'}(ws) \leq t_{j',D'}(\text{update}^*(w)s) \text{ and } s_{j',D'}(ws) \leq s_{j',D'}(\text{update}^*(w)s).$$

There are two possibilities for $\text{update}^*(ws)$. If $t_{j,D}(\text{update}^*(w)s) \leq f_j^{-1}(b_P(\mathcal{G}))$, then

$$\text{update}^*(ws) = \text{update}(\text{update}^*(w), s) = \text{update}^*(w)s.$$

Thus,

$$t_{j',D'}(ws) \leq t_{j',D'}(\text{update}^*(w)s) = t_{j',D'}\text{update}^*(ws)$$

and

$$s_{j',D'}(ws) \leq s_{j',D'}(\text{update}^*(w)s) = s_{j',D'}\text{update}^*(ws).$$

If $t_{j,D}(\text{update}^*(w)s) > f_j^{-1}(b_P(\mathcal{G}))$, then

$$\text{update}^*(ws) = \text{update}(\text{update}^*(w), s) = zs$$

where $t_{j',D'}(zs) \geq t_{j',D'}(\text{update}^*(w)s)$ and $s_{j',D'}(zs) \geq s_{j',D'}(\text{update}^*(w)s)$ by definition of $I_{j,D}$. Hence,

$$t_{j',D'}(ws) \leq t_{j',D'}(\text{update}^*(w)s) \leq t_{j',D'}(zs) = t_{j',D'}(\text{update}^*(ws))$$

and

$$s_{j',D'}(ws) \leq s_{j',D'}(\text{update}^*(w)s) \leq s_{j',D'}(zs) = s_{j',D'}(\text{update}^*(ws))$$

for all $j' \in [k]$ and all $D' \in \text{Up}(\mathcal{P}_{j'})$. □

The following results are again easy implications of Lemma 4.16. They conclude the discussion of the strategy improvement operator for Poset Games.

Lemma 4.17. *Let σ be a winning strategy for Player 0 for $\mathcal{G}$, $j \in [k]$, and $D \in \text{Up}(\mathcal{P}_j)$.*

(i) If σ bounds the totalized waiting time for some $D' \in \text{Up}(\mathcal{P}_j)$ to B, then so does $I_{j,D}(\sigma)$.

(ii) $v_P(I_{j,D}(\sigma)) \leq v_P(\sigma)$.

(iii) $I_{j,D}(\sigma)$ is a winning strategy for Player 0, if σ is a winning strategy for her.

We now explain how to improve a given strategy by applying the strategy improvement operators. Every interval in which a request is continuously open is divided into several smaller intervals by the positions into which an element of the poset is embedded. The order of improvement reduces the early subintervals, those with *large* D, first and then the later ones, with *small* D.

Example 4.18. Consider the play ρ, depicted in Figure 4.5, of a game with a single condition $(q, (D, \preccurlyeq))$ where $D = \{d_1, d_2, d_3\}$ and $d_1 \preccurlyeq^{red} d_2 \preccurlyeq^{red} d_3$ (again, we ignore the labeling functions). The improvement scheme to be defined starts with the improvement with respect to D, then with respect to $\{d_2, d_3\}$ and finally with respect to $\{d_3\}$.

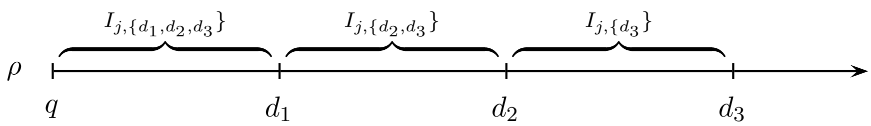

Figure 4.5: The order of improvement with $I_{j,D}$ for $D \subseteq \{d_1, d_2, d_3\}$

Thus, if the strategy is improved with respect to some $D \in \text{Up}(\mathcal{P}_j)$, then we are sure that the earlier subintervals are already improved and do not have to be changed (for condition j). The order of improvement is given by an enumeration of the upwards-closed subsets of D_j for every j: for $j \in [k]$, let $c_j = |\text{Up}(\mathcal{P}_j)|$ and $e_j : [c_j] \to \text{Up}(\mathcal{P}_j)$ be an enumeration such that $|D| > |D'|$ implies $e_j^{-1}(D) < e_j^{-1}(D')$. Thus, the subsets are enumerated in order of decreasing size. The single step operator deletes loops of every long interval in which no element of the domain is embedded. However, the residues of the interval may form new loops that have to be deleted. Therefore, $I_{j,D}$ is applied infinitely often for every $D \in \mathrm{Up}(\mathcal{P}_j)$ of every condition j.

Given a winning strategy σ_0 for Player 0 such that $v_P(\sigma_0) \leq b_P(\mathcal{G})$, we define

- $\sigma_{j,l,0} = \begin{cases} \sigma_{j-1} & \text{if } l = 1 \\ \sigma_{j,l-1} & \text{otherwise} \end{cases}$ for $j \in [k]$ and $l \in [c_j]$,
- $\sigma_{j,l,n+1} = I_{j,e_j(l)}(\sigma_{j,l,n})$ for $j \in [k]$, $l \in [c_j]$, and $n \in \mathbb{N}$,
- $\sigma_{j,l} = \lim_{n\to\infty} \sigma_{j,l,n}$ for $j \in [k]$ and $l \in [c_j]$, and
- $\sigma_j = \sigma_{j,c_j}$ for $j \in [k]$.

For notational convenience, we define $\sigma_{j,0} = \sigma_{j-1}$ for $j \in [k]$. The improvement scheme is visualized in Figure 4.6.

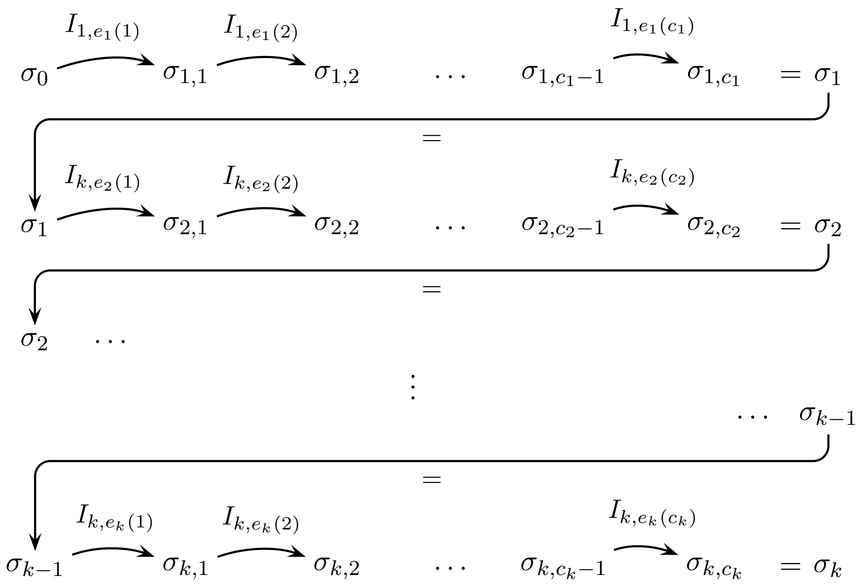

Figure 4.6: The improvement scheme for a Poset Game

It remains to lift the lemmata stated above for a single improvement step to the limit of the improvement steps and to show that σ_k uniformly bounds the waiting time for all $j \in [k]$ and all $D \in \mathrm{Up}(\mathcal{P}_j)$. Then, σ_k also uniformly bounds the totalized waiting time for all $j \in [k]$ and all $D \in \mathrm{Up}(\mathcal{P}_j)$ by Lemma 4.11.

The strategy improvement operator skips loops if the values $t_{j,D}(w)$ and $s_{j,D}(w)$ (for every D) are higher at the end of the loop than they were at the beginning. To obtain the desired bounds we again determine the maximal length of a non-dickson sequence

[12] of vectors containing these values, which can only increase, be reset to zero or stay zero. This results in a bound that depends only on the number of vertices of the arena and the number of upwards-closed subsets.

Let w_1 and w_2 be finite plays such that $w_1 \sqsubset w_2$, and let $D \in \text{Up}(\mathcal{P}_j)$. We say that w_1 and w_2 are *dickson save* D iff the last vertices of w_1 and w_2 are equal and $t_{j',D'}(w_1) \leq t_{j',D'}(w_2)$ and $s_{j',D'}(w_1) \leq s_{j',D'}(w_2)$ for all j' and all $D' \in \text{Up}(\mathcal{P}_{j'})$ such that $D \neq D'$. If w_1 and w_2 are dickson save D, the totalized waiting time for D is high enough and strictly increasing in between w_1 and w_2, then the strategy improvement operator $I_{j,D}$ deletes the loop $(w_1)^{-1}w_2$.

Now, let xy be a finite play. We say that y is *non-dickson save* D if xy does not have a pair of dickson prefixes that both do contain a non-empty part of y. The strategy improvement operator $I_{j,D}$ does not delete any loops in a prefix that is non-dickson save D. We bound the length of a non-dickson sequence in terms of the size of the arena and the number of upwards-closed subsets by a function b.

If there is only one subset, namely D, then there must not be a state repetition in y. Thus, the length of a non-dickson sequence can be bounded by the number of vertices in $|G|$; hence, we define $b(m, 1) := m$. Now, assume there are $k + 1$ subsets. If $t_{j',D'}$ is non-decreasing for some subset D' with $D' \neq D$ for more than $b(|G|, k)$ steps (which implies that $s_{j',D'}$ is non-decreasing as well), then there are two dickson prefixes, with respect to the other k subsets in that infix that are also dickson for all $k + 1$ subsets. Hence, in every non-dickson sequence the value $t_{j',D'}$ for every subset $D' \neq D$ has to be reset to 0 in every infix of length $b(|G|, k)$. If $t_{j',D'}(x) = 0$, then we can bound $t_{j',D'}(xy)$ and $s_{j',D'}(xy)$ by

$$\frac{|y| \cdot (|y| + 1)}{2}.$$

Altogether, the values in a non-dickson sequence are bounded by

$$\frac{b(|G|, k) \cdot (b(|G|, k) + 1)}{2}$$

after an initial prefix of length $b(|G|, k)$. Thus, there cannot be more than

$$b(|G|, k) + |G| \cdot \left(\frac{b(|G|, k) \cdot (b(|G|, k) + 1)}{2}\right)^{2k}$$

steps without a pair of dickson prefixes, which gives

$$b(m, k + 1) := b(m, k) + m \cdot \left(\frac{b(m, k) \cdot (b(m, k) + 1)}{2}\right)^{2k}.$$

The exponent $2k$ is due to the fact that the vectors contain two value for each subset $D' \in \text{Up}(\mathcal{P}_{j'})$, i.e., $t_{j',D'}(w)$ and $s_{j',D'}(w)$.

Now, we state the last technical Lemma which sums up all properties of the improvement scheme we need. Afterwards, we conclude the first step by putting the pieces together.

Lemma 4.19. *Let $j \in [k]$, $l \in [c_j]$, and $e(l) = (j, D)$. Then*

(i) $(\text{update}^*_{j,l,n})_{n\in\mathbb{N}}$ *converges to the identity function.*

(ii) $\lim_{n\to\infty} \sigma_{j,l,n}$ *exists.*

(iii) If $\sigma_{j,l-1}$ uniformly bounds the totalized waiting time for $D' \in \text{Up}(\mathcal{P}_{j'})$ for some j', then so does $\sigma_{j,l}$.

(iv) $v_P(\sigma_{j,l}) \leq v_P(\sigma_{j,l-1})$.

(v) $\sigma_{j,l}$ uniformly bounds the waiting time for D to

$$b_{j,D} := f_j^{-1}(b_P(\mathcal{G})) + (|D_j \backslash D| + 1) \cdot b\left(|G|, \sum_{j=1}^{k} |\text{Up}(\mathcal{P}_j)|\right).$$

Proof. (i) We proceed by induction: we have $\text{update}^*_{j,l,n}(s_0) = s_0$ for all n. Now, assume $\text{update}^*_{j,l,n}(w) = w$ for all $n \geq n_w$. If $t_{j,D}(ws) \leq f_j^{-1}(b_P(\mathcal{G}))$, then $\text{update}_{j,l,n}(w, s) = ws$ and therefore also $\text{update}^*_{j,l,n}(ws) = ws$ for all $n \geq n_w$.

Thus, let $t_j(ws) > f_j^{-1}(b_P(\mathcal{G}))$ and let $\mathfrak{T}_n$ be $\mathfrak{T}^{\sigma_{j,l,n-1}}_{G,s_0}\upharpoonright_{ws}$ restricted to the maximal paths that continuously contain D in their set of open requests. By definition, $\text{update}^*_{j,l,n}(ws)$ is a vertex of $\mathfrak{T}_n$. Since every path in $\mathfrak{T}_n$ is a path in $\mathfrak{T}_{n-1}$ from which some loops might be deleted the size of the $\mathfrak{T}_n$ is decreasing. Finally, if $\mathfrak{T}_n = \mathfrak{T}_{n+1}$, then we have $\mathfrak{T}_{n'} = \mathfrak{T}_n$ for all $n' \geq n$. Thus, there is an index $n_{ws} \geq n_w$ such that the $\mathfrak{T}_n$ are equal for all $n \geq n_{ws}$. From that index on we have $\text{update}_{j,l,n}(w, s) = ws$ and thus $\text{update}^*_{j,l,n}(ws) = ws$.

(ii) By (i), we have

$$\begin{aligned}
&\sigma_{j,l,n}(\rho_0 \dots \rho_i) \\
=&\text{next}_{j,l,n}(\rho_i, \text{update}^*_{j,n}(\rho_0 \dots \rho_i)) \\
=&\text{next}_{j,l,n}(\rho_i, \rho_0 \dots \rho_i) \\
=&\sigma_{j,l,n-1}(\rho_0 \dots \rho_i)
\end{aligned}$$

for all finite plays $\rho_0 \dots \rho_i$ and all sufficiently large n. Thus, $(\sigma_{j,l,n})_{n\in\mathbb{N}}$ converges.

For the next two claims we need to introduce some additional but familiar notation. For an explanation we refer to Figure 3.3. For $1 \leq m \leq n$ define $\text{update}^*_{j,l,[m,n]}$ by $\text{update}^*_{j,l,[m,m]}(w) = \text{update}^*_{j,l,m}(w)$ and

$$\text{update}^*_{j,l,[m,n+1]}(w) = \text{update}^*_{j,l,[m,n]}(\text{update}^*_{j,l,n+1}(w)).$$

Applying Lemma 4.15 inductively, we can show that $\text{update}^*_{j,l,[m,n]}(w)$ is a finite play consistent with $\sigma_{j,l,m-1}$ for every play w consistent with $\sigma_{j,l,n}$. Analogously, applying Lemma 4.16 inductively, we get $t(w) \leq t(\text{update}^*_{j,l,[m,n]}(w))$ for all w consistent with σ_n.

We have shown in the proof of Lemma 4.19 (i) that for every finite play x according to $\sigma_{j,l}$ there is an n_x such that $\text{update}^*_{j,l,n}(x) = x$ for all $n \geq n_x$. We define $\text{update}^*_{j,l,\omega}$ by $\text{update}_{j,l,\omega}(x) = \text{update}^*_{j,l,[1,n_x]}(x)$. By the remarks above, we know that $\text{update}^*_{j,l,\omega}(x)$ is consistent with $\sigma_{j,l-1}$ and $t(x) \leq t(\text{update}^*_{j,l,\omega}(x))$ for every play x consistent with $\sigma_{j,l}$.

(iii) We have $t_{j',D'}(x) \leq B$ for all finite plays x consistent with $\sigma_{j,l-1}$. Now, let x be a play consistent with $\sigma_{j,l}$. Then, $\text{update}^*_{j,l,\omega}(x)$ is a prefix of a play according to $\sigma_{j,l-1}$. Hence, $t_{j',D'}(x) \leq t_{j',D'}(\text{update}^*_{j,l,\omega}(x)) \leq B$. Thus, $\sigma_{j,l}$ uniformly bounds the totalized waiting time for D'.

(iv) For a play $\rho = \rho_0\rho_1\rho_2\ldots$, let $\text{update}^*_{j,l,\omega}(\rho) = \lim_{n\to\infty} \text{update}^*_{j,l,\omega}(\rho_0 \ldots \rho_n)$. The limit $\text{update}^*_{j,l,\omega}(\rho)$ is a play consistent with $\sigma_{j,l-1}$ for every play ρ consistent with $\sigma_{j,l}$. We show $v_P(\rho) \leq v_P(\text{update}^*_{j,\omega}(\rho))$ for all ρ consistent with $\sigma_{j,l}$, which implies the claim. To this end, we define

$$S = \{x' \sqsubset \text{update}^*_{j,l,\omega}(\rho) \mid \neg\exists x \sqsubset \rho : \text{update}^*_{j,l,\omega}(x) = x'\}.$$

S contains exactly the vertices of the loops skipped by Player 0 throughout the improvement steps. Let $x' \in S$; then, $t_j(x') > f_j^{-1}(b_P(\mathcal{G}))$ holds, as every improvement step only deletes loops of $\text{update}^*_{j,l,\omega}(\rho)$ after a waiting time of at least $f_j^{-1}(b_P(\mathcal{G}))$ steps. Thus, $p(x') > b_P(\mathcal{G}) \geq v_P(\sigma)$ and

$$\limsup_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} p(\text{update}^*_{j,l}(\rho_0 \ldots \rho_i)) \leq v_P(\text{update}^*_{j,l}(\rho)) \tag{4.1}$$

since the average decreases if the summation omits the summands for the prefixes in S. Now, let $x \sqsubset \rho$: We have $t(x) \leq t(\text{update}^*_{j,l,\omega}(x))$ and therefore $p(x) \leq p(\text{update}^*_{j,l,\omega}(x))$. Thus,

$$\frac{1}{n} \sum_{i=0}^{n-1} p(\rho_0 \ldots \rho_i) \leq \frac{1}{n} \sum_{i=0}^{n-1} p(\text{update}^*_{j,l}(\rho_0 \ldots \rho_i)).$$

The latter term converges to a value less than or equal to $v_P(\text{update}^*_{j,l}(\rho))$, by (4.1). Thus, we conclude $v_P(\rho) \leq v_P(\text{update}^*_{j,l}(\rho))$.

(v) The last claim is proven by induction over l. There are at most $|D_j \backslash D|$ intervals in between the elements of D_j that were embedded already. By induction hypothesis, we can assume that the claim holds for all those intervals, i.e., for all $D' \in \text{Up}(\mathcal{P}_j)$ such that $|D'| > |D|$, since $e^{-1}(j, D') < e^{-1}(j, D)$. Now, assume there is a play w consistent with $\sigma_{j,l}$ such that $(D, t) \in \text{Open}_j(w)$ for some $t > b_{j,D}$ and let $w = xy$ such that

$|y| = d(|G|, \sum_{j=1}^{k} |\mathrm{Up}(\mathcal{P}_j)|) + 1$. From the induction hypothesis and the fact that the waiting times for all D' that could precede D are already bounded, we can conclude that

$$(D, f_j^{-1}(b_P(\mathcal{G})) + (|D \setminus D_j|) \cdot b(|G|, \sum_{j=1}^{k} |\mathrm{Up}(\mathcal{P}_j)|) + 1 + |y'|) \in \mathrm{Open}_j(xy')$$

for all $y' \sqsubseteq y$. The situation is depicted in Figure 4.7: the length of the first two intervals I is bounded by induction hypothesis. Thus, if the waiting time is longer than $b_{j,D}$, then the last interval is longer than $b(|G|, \sum_{j=1}^{k}(|\mathrm{Up}(\mathcal{P}_j)|))$.

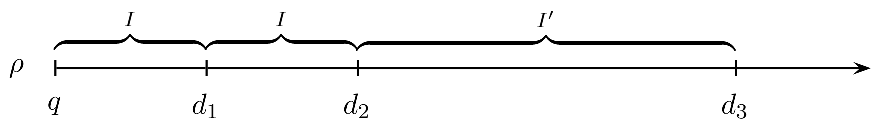

Figure 4.7: The inductive step for Lemma 4.19 (v): the intervals I are *short* by induction hypothesis. Thus, I' is *long*

Finally, there exists N such that $\mathrm{update}^*_{j,l,N}(xy') = xy'$ for all $y' \sqsubseteq y$. Thus, y is non-dickson save D, which contradicts the definition of d. □

Now, we can wrap things up by stating the final lemma of the first part, which gives us the desired bound on the waiting times of an optimal strategy.

Lemma 4.20. *For every winning strategy σ_0 for $\mathcal{G}$ for Player 0 of value $v_P(\sigma_0) \leq b_P(\mathcal{G})$, there is a winning strategy σ_k that bounds the totalized waiting time for all $j \in [k]$ and all $D \in \mathrm{Up}(\mathcal{P}_j)$ to $\frac{b_{j,D} \cdot (b_{j,D}+1)}{2}$. Furthermore, $v_P(\sigma_k) \leq v_P(\sigma_0)$.*

Proof. By induction over l. From Lemma 4.19 (iii) and (v) and Remark 4.11 we conclude that σ_k bounds the totalized waiting time for all $j \in [k]$ and all $D \in \mathrm{Up}(\mathcal{P}_j)$ to $\frac{b_{j,D} \cdot (b_{j,D}+1)}{2}$. Analogously, $v_P(\sigma_k) \leq v_P(\sigma_0) \leq b_P(\mathcal{G})$ by induction hypothesis, by Lemma 4.19 (iv), and by the assumption on σ_0. Thus, σ_k is a winning strategy by Lemma 4.10 (ii). □

This concludes the first step. We have proved that the search for an optimal strategy can be restricted to a finite domain. In the second step, we construct a Mean-Payoff Game whose plays are the plays of $\mathcal{G}$ annotated with their totalized waiting times. Lemma 4.20 allows us to bound the totalized waiting time for $D \in \mathrm{Up}(\mathcal{P}_j)$ to $\frac{b_{j,D} \cdot (b_{j,D}+1)}{2}$, and thereby also the size of the expanded arena. Then, we find an optimal strategy for the Poset Game by determining the optimal strategy in a Mean-Payoff Game.

4.3.2 Reducing Poset Games to Mean-Payoff Games

This subsection is devoted to the second step of the proof of Theorem 4.13, the reduction to Mean-Payoff Games. In the expanded arena, we need to keep track of the totalized waiting time $t_{j,D}$. To be able to compute $t_{j,D}(ws)$ from $t_{j,D}(w)$ locally we need to know $s_{j,D}(w)$ as well. The bound on $t_{j,D}(w)$ obtained in the first part of the proof also bounds $s_{j,D}(w)$. Thus, let F_j be the set of functions

$$f : \mathrm{Up}(\mathcal{P}_j) \to \mathbb{N} \quad \text{such that} \quad f(D) \leq \frac{b_{j,D} \cdot (b_{j,D} + 1)}{2}$$

for all $D \in \mathrm{Up}(\mathcal{P}_j)$. Every F_j is obviously finite.

Remember that $\mathrm{New}_j(s)$ returns D_j if s is a request for condition j, and $\emptyset$ otherwise. Furthermore, $\mathrm{Emb}_j(D, s)$ is the set of elements in D that could not be embedded to the vertex s. Formally,

$$\mathrm{New}_j(s) = \begin{cases} D_j & \text{if } q_j \in l_G(s) \\ \emptyset & \text{otherwise} \end{cases}$$

and

$$\mathrm{Emb}_j(D, s) = \{d \in D \mid \exists d' \in D : d' \preccurlyeq_j d \text{ and } l_j(d') \notin l_G(s)\}.$$

We define the memory structure $\mathfrak{M} = (M, \mathrm{init}, \mathrm{update})$ where

- $M = \prod_{j=1}^{k} (F_j \times F_j) \cup \{m_\uparrow\}$,
- $\mathrm{init}(s) = (\mathrm{time}_1, \mathrm{size}_1, \dots, \mathrm{time}_k, \mathrm{size}_k)$ where

$$\mathrm{time}_j(D) = \mathrm{size}_j(D) = \begin{cases} 1 & \text{if } \mathrm{Emb}_j(\mathrm{New}_j(s), s) = D \\ 0 & \text{otherwise} \end{cases}$$

  for all $D \in \mathrm{Up}(\mathcal{P}_j)$, and

- The update function is given by $\mathrm{update}(m_\uparrow, s) = m_\uparrow$ for all vertices s. Otherwise, if $m = (\mathrm{time}_1, \mathrm{size}_1, \dots, \mathrm{time}_k, \mathrm{size}_k) \in M$ define time'_j and size'_j by

$$\mathrm{time}'_j(D) = |\mathrm{Emb}_j(\mathrm{New}_j(s), s) \cap \{D\}| + \sum_{\substack{D' \in \mathrm{Up}(\mathcal{P}_j): \\ \mathrm{Emb}_j(D', s) = D}} (\mathrm{time}_j(D') + \mathrm{size}_j(D'))$$

  and

$$\mathrm{size}'_j(D) = |\mathrm{Emb}_j(\mathrm{New}_j(s), s) \cap \{D\}| + \sum_{\substack{D' \in \mathrm{Up}(\mathcal{P}_j): \\ \mathrm{Emb}_j(D', s) = D}} \mathrm{size}_j(D')$$

for all $D \in \mathrm{Up}(\mathcal{P}_j)$. If $\mathrm{time}'_j \notin F_j$ for some j, i.e., there exists $D \in \mathrm{Up}(\mathcal{P}_j)$ such that $\mathrm{time}'_j(D) > \frac{b_{j,D} \cdot (b_{j,D}+1)}{2}$, then we define $\mathrm{update}(m, s) = m_\uparrow$, otherwise $\mathrm{update}(m, s) = (\mathrm{time}'_1, \mathrm{size}'_1, \ldots, \mathrm{time}'_k, \mathrm{size}'_k)$.

Since it is Player 1's goal to minimize the limit superior of the average edge weights in a Mean-Payoff Game we have to switch the Player's positions, i.e., we define $V'_0 = V_1 \times M$, $V'_1 = V_0 \times M$ and $G' = (V \times M, V'_0, V'_1, E_{\mathrm{update}})$. Let

$$d = \sum_{j=1}^{k} f_j\Big(\frac{b_{j,D} \cdot (b_{j,D}+1)}{2} + 1\Big)$$

where f_j is the penalty function for condition j. To complete the definition of the game we need to define the weight function l: We define $l((s, m), (s', m_\uparrow)) = d$ for all $(s, m) \in V \times M$ and

$$\begin{aligned} &l((s, (\mathrm{time}_1, \mathrm{size}_1, \ldots, \mathrm{time}_k, \mathrm{size}_k)), (s', (\mathrm{time}'_1, \mathrm{size}'_1, \ldots, \mathrm{time}'_k, \mathrm{size}'_k))) \\ =&\sum_{j=1}^{k} f_j\left(\sum_{D \in \mathrm{Up}(\mathcal{P}_j)} \mathrm{time}_j(D)\right) \end{aligned}$$

where f_j is again the penalty function. Now, let $\mathcal{G}' = (G', (s_0, \mathrm{init}(s_0)), d, l)$ be the Mean-Payoff Game in the expanded arena.

The following lemma shows that the values for a play $\mathcal{G}$ and the expanded plays in $\mathcal{G}'$ are equal.

Lemma 4.21. *Let w be a finite play of G.*

(i) *Let* $\mathrm{update}(w) = (\mathrm{time}_1, \mathrm{size}_1, \ldots, \mathrm{time}_k, \mathrm{size}_k)$. *Then,* $\mathrm{time}_j(D) = t_{j,D}(w)$ *and* $\mathrm{size}_j(D) = s_{j,D}(w)$.

(ii) *If* $\mathrm{update}(w) = m_\uparrow$, *then there exists a prefix w' of w and a $D \in \mathrm{Up}(\mathcal{P}_j)$ such that* $t_{j,D}(w') > \frac{b_{j,D} \cdot (b_{j,D}+1)}{2}$.

(iii) *Let ρ be a play in G such that the totalized waiting times for all $j \in [k]$ and all $D \in \mathrm{Up}(\mathcal{P}_j)$ are uniformly bounded by $\frac{b_{j,D} \cdot (b_{j,D}+1)}{2}$, and let ρ' be the expanded play in G'. Then, $v_P(\rho) = v_1(\rho')$.*

(iv) *Let ρ' be a play in G' that does not visit a vertex with memory state $m_\uparrow$, and let ρ be the projected play in G. Then, $v_1(\rho') = v_P(\rho)$.*

Proof. (i) By induction over w: for the base case $w = s$ notice that $t_{j,D}(s) \in \{0, 1\}$. We have

$$\begin{aligned} t_{j,D}(s) = 1 \Leftrightarrow\ & (D, 1) \in \mathrm{Open}_j(s) \\ \Leftrightarrow\ & D = \mathrm{Emb}_j(\mathrm{New}_j(s), s) \Leftrightarrow\ \mathrm{time}_j(D) = 1, \end{aligned}$$

and

$$\begin{aligned} t_{j,D}(s) = 0 &\Leftrightarrow\ (D,1) \notin \mathrm{Open}_j(s) \\ &\Leftrightarrow\ D \neq \mathrm{Emb}_j(\mathrm{New}_j(s), s) \Leftrightarrow\ \mathrm{time}_j(D) = 0. \end{aligned}$$

Also, $s_{j,D}(s) \in \{0,1\}$ and

$$\begin{aligned} s_{j,D}(w) = 1 &\Leftrightarrow\ \mathrm{Open}_j(s) \cap \{D\} \times \mathbb{N} = \{(D,1)\} \\ &\Leftrightarrow\ t_{j,D} = 1 \Leftrightarrow\ \mathrm{time}_j(D) = 1 \Leftrightarrow\ \mathrm{size}_j(D) = 1, \end{aligned}$$

and

$$\begin{aligned} s_{j,D}(w) = 0 &\Leftrightarrow\ \mathrm{Open}_j(s) \cap \{D\} \times \mathbb{N} = \emptyset \\ &\Leftrightarrow\ t_{j,D} = 0 \Leftrightarrow\ \mathrm{time}_j(D) = 0 \Leftrightarrow\ \mathrm{size}_j(D) = 0. \end{aligned}$$

For the induction step, let $\mathrm{update}^*(w) = (\mathrm{time}_1, \mathrm{size}_1, \ldots, \mathrm{time}_k, \mathrm{size}_k)$ and similarly $\mathrm{update}^*(ws) = (\mathrm{time}'_1, \mathrm{size}'_1, \ldots, \mathrm{time}'_k, \mathrm{size}'_k)$. By induction hypothesis, we can assume $\mathrm{time}_j(D) = t_{j,D}(w)$ and $\mathrm{size}_j(D) = s_{j,D}(w)$. We have

$$\begin{aligned} & t_{j,D}(ws) \\ =& \sum_{(D,t)\in \mathrm{Open}_j(ws)} t \\ =& |\mathrm{Emb}_j(\mathrm{New}_j(s), s) \cap \{D\}| + \sum_{\substack{(D',t')\in \mathrm{Open}_j(w): \\ \mathrm{Emb}_j(D',s)=D}} (t'+1) \\ =& |\mathrm{Emb}_j(\mathrm{New}_j(s), s) \cap \{D\}| + \sum_{\substack{(D',t')\in \mathrm{Up}(\mathcal{P}_j): \\ \mathrm{Emb}_j(D',s)=D}} (t_{j,D'}(w) + s_{j,D'}(w)) \\ =& |\mathrm{Emb}_j(\mathrm{New}_j(s), s) \cap \{D\}| + \sum_{\substack{D'\in \mathrm{Up}(\mathcal{P}_j): \\ \mathrm{Emb}_j(D',s)=D}} (\mathrm{time}_j(D') + \mathrm{size}_j(D')) \\ =& \mathrm{time}'_j(D), \end{aligned}$$

and

$$\begin{aligned} & s_{j,D}(ws) \\ =& |\{t \mid (D,t) \in \mathrm{Open}_j(ws)\}| \\ =& |\{\mathrm{Emb}_j(\mathrm{New}_j(s), s) \cap \{D\}\}| + \sum_{\substack{D'\in \mathrm{Up}(\mathcal{P}_j): \\ \mathrm{Emb}(D',s)=D}} |\{t' \mid (D',t') \in \mathrm{Open}_j(w)\}| \end{aligned}$$

$$=|\{\mathrm{Emb}_j(\mathrm{New}_j(s), s) \cap \{D\}\}| + \sum_{\substack{D' \in \mathrm{Up}(\mathcal{P}_j): \\ \mathrm{Emb}(D', s) = D}} s_{j,D'}(w)$$

$$=|\mathrm{Emb}_j(\mathrm{New}_j(s), s) \cap \{D\}| + \sum_{\substack{D' \in \mathrm{Up}(\mathcal{P}_j): \\ \mathrm{Emb}_j(D', s) = D}} \mathrm{size}_j(D')$$

$$=\mathrm{size}'_j(D).$$

(ii) Let w' be the longest prefix of w such that $\mathrm{update}^*(w') \neq m_\uparrow$ and s the next vertex of w after w'. Furthermore, let $\mathrm{update}^*(w') = (\mathrm{time}_1, \mathrm{size}_1, \dots, \mathrm{time}_k, \mathrm{size}_k)$. Thus, $\mathrm{time}_j(D) = t_{j,D}(w)$ by (i). We define time'_j and size'_j by

$$\mathrm{time}'_j(D) = |\mathrm{Emb}_j(\mathrm{New}_j(s), s) \cap \{D\}| + \sum_{\substack{D' \in \mathrm{Up}(\mathcal{P}_j): \\ \mathrm{Emb}_j(D', s) = D}} (\mathrm{time}_j(D') + \mathrm{size}_j(D'))$$

and

$$\mathrm{size}'_j(D) = |\mathrm{Emb}_j(\mathrm{New}_j(s), s) \cap \{D\}| + \sum_{\substack{D' \in \mathrm{Up}(\mathcal{P}_j): \\ \mathrm{Emb}_j(D', s) = D}} \mathrm{size}_j(D'),$$

respectively. Analogous to the proof in (i), we can show that $\mathrm{time}'_j(D) = t_{j,D}(ws)$ and $size'_j(D) = s_{j,D}(ws)$. Assume $t_{j,D}(w') \leq \frac{b_{j,D} \cdot (b_{j,D}+1)}{2}$ for all $D \in \mathrm{Up}(\mathcal{P}_j)$ and all $j \in [k]$. Then, $\mathrm{update}((\mathrm{time}_1, \mathrm{size}_1, \dots, \mathrm{time}_k, \mathrm{size}_k), s) = \mathrm{update}^*(w's) \neq m_\uparrow$ which contradicts our assumption on w'. Thus, $t_{j,D}(w') > \frac{b_{j,D} \cdot (b_{j,D}+1)}{2}$ for some $D \in \mathrm{Up}(\mathcal{P}_j)$ and some j.

(iii) and (iv) are easy implications of (i) and (ii). □

Now, we are able to prove Theorem 4.13.

Proof. We begin by relating strategies and values for Player 0 for $\mathcal{G}$ and Player 1 in $\mathcal{G}'$.

Let σ be a strategy for Player 0 for $\mathcal{G}$ that uniformly bounds the totalized waiting times for all conditions $D \in \mathrm{Up}(\mathcal{P}_j)$ to $\frac{b_{j,D} \cdot (b_{j,D}+1)}{2}$. We define the strategy τ' for Player 1 in $\mathcal{G}'$ by $\tau'((\rho_0, m_0) \dots (\rho_n, m_n)) = \sigma(\rho_0 \dots \rho_n)$. We claim τ' guarantees $v_P(\sigma)$ for Player 1 in $\mathcal{G}'$. Assume it does not. Then, Player 0 has a strategy σ' for $\mathcal{G}'$ such that $v_1(\rho(s_0, \sigma', \tau')) > v_P(\sigma)$. The projected play ρ of $\rho(s_0, \sigma', \tau')$ is consistent with σ by construction of τ'. Thus, $v_P(\sigma) \geq v_P(\rho) = v_1(\rho(s_0, \sigma', \tau')) > v_P(\sigma)$ by Lemma 4.21 (iii), which yields the desired contradiction.

Conversely, let τ' be a strategy for Player 1 in $\mathcal{G}'$ that guarantees a loss $d' < d$. Thus, no play consistent with τ' visits a vertex with memory state $m_\uparrow$. Let σ be the strategy for Player 0 for $\mathcal{G}$ induced by τ' via $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$. We claim $v_P(\sigma) \leq d'$. Assume Player 1 has a strategy τ for $\mathcal{G}$ such that $v_P(\rho(s_0, \sigma, \tau)) > d'$. The expanded play ρ' of

$\rho(s_0, \sigma, \tau)$ is consistent with τ' by Lemma 2.8. Thus, $d' \geq v_1(\rho') = v_P(\rho(s_0, \sigma, \tau)) > d'$ by Lemma 4.21 (iv), which again amounts to a contradiction.

Now, we can begin with the actual proof: since Player 0 wins $\mathcal{G}$, Corollary 4.12 and Lemma 4.20 guarantee that she also has a winning strategy σ that uniformly bounds the waiting times for all $D \in \mathrm{Up}(\mathcal{P}_j)$ to $\frac{b_{j,D} \cdot (b_{j,D}+1)}{2}$. Let τ' be the induced strategy for Player 1 in $\mathcal{G}'$. Every play consistent with τ' does not reach a vertex with memory state $m_\uparrow$. Thus, this strategy guarantees a loss less than d. Hence, $v_M(\mathcal{G}') < d$.

Let τ_{opt} be a positional strategy guaranteeing $v_M(\mathcal{G}')$ for Player 1 in $\mathcal{G}'$. We show that the strategy σ_{opt} induced by $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$ and τ_{opt} is an optimal winning strategy for Player 0 for $\mathcal{G}$. This suffices since σ_{opt} is finite-state with memory $\mathfrak{M}$ and effectively computable by Theorem 2.20.

By the remarks above, we have $v_P(\sigma_{opt}) = v_M(\mathcal{G}')$. To conclude the proof we assume that σ_{opt} is not optimal, i.e., Player 0 has a strategy σ for $\mathcal{G}$ such that $v_P(\sigma) < v_P(\sigma_{opt})$. By Lemma 4.20, we can assume without loss of generality that σ uniformly bounds the totalized waiting times for all $D \in \mathrm{Up}(\mathcal{P}_j)$ to $\frac{b_{j,D} \cdot (b_{j,D}+1)}{2}$. Then, the strategy τ' for Player 1 in $\mathcal{G}'$ induced by σ guarantees $v_P(\sigma) < v_P(\sigma_{opt}) = v_M(\mathcal{G}')$. This amounts to a contradiction, since τ_{opt} is optimal for $\mathcal{G}'$. □

Corollary 4.22. *The value $v_P(\sigma)$ of an optimal strategy σ is effectively computable.*

Proof. Construct $\mathcal{G}'$ and compute $v_M(\mathcal{G}')$. If $v_M(\mathcal{G}') = d$, then Player 0 loses $\mathcal{G}$ and $v(\sigma) = \infty$ for every strategy σ. Otherwise, the value of an optimal strategy for the Poset Games coincides with $v_M(\mathcal{G}')$. □

Every Request-Response Game is a Poset Game where every domain is a singleton. Formally, given a Request-Response Game $\mathcal{G} = (G, s_0, (Q_j, P_j)_{j=1,\ldots,k})$ construct the Poset Game $\mathcal{G}' = (G, s_0, (q_j, \mathcal{P}_j)_{j=1,\ldots,k})$ where $\mathcal{P}_j = (\{p_j\}, \{(p_j, p_j)\}, l_j)$ and $l_j(p_j) = p_j$. The labeling l_G of the arena G is defined by $l_G(s) = \{q_j \mid s \in Q_j\} \cup \{p_j \mid s \in P_j\}$. The games $\mathcal{G}$ and $\mathcal{G}'$ are obviously equivalent.

Hence, both frameworks for defining time-optimal strategies are applicable to $\mathcal{G}$. Either use the framework for Request-Response Games or construct $\mathcal{G}'$ first, and then use the framework for Poset Games. We have already seen that the waiting times for Request-Response Games are defined by a single clock for every condition that is started by request and stopped by the first response. All subsequent requests of a condition that occur while the clock is still running are ignored. Waiting times for Poset Games have to be defined differently since the embeddings for different requests may overlap (see Figure 4.3). Thus, for every request of a condition a new clock is started and is not stopped until the corresponding poset is embedded. Thus, in a Poset Game $\mathcal{G}'$ constructed from a Request-Response Game, requests occurring while another request is open, are not ignored. Therefore, the value of a play or a strategy for Player 0, respectively, is smaller in the Request-Response Game setting than in the Poset Game setting, if the values are defined with respect to the same penalty functions.

Lemma 4.23. *Let ρ be a play and σ a strategy for Player 0.*

(i) $v_R(\rho) \leq v_P(\rho)$.

(ii) $v_R(\sigma) \leq v_P(\sigma)$.

Proof. (i) We prove $(\{p_j\}, t_j(\rho_0 \dots \rho_n)) \in \mathrm{Open}_j(\rho_0 \dots \rho_n)$, if $t_j(\rho_0 \dots \rho_n) > 0$, which implies the claim. Here, t_j is the waiting time function for the Request-Response Game. Let $t_j(\rho_0 \dots \rho_n) = 1$. Then, $\rho_n \in Q_j \backslash P_j$ by definition of t_j. Therefore, $q_j \in l_G(s)$ but $p_j \notin l_G(s)$, by definition of l_G. Hence, it holds $(\{p_j\}, 1) \in \mathrm{Open}_j(\rho_0)$. Now, let $t_j(\rho_0 \dots \rho_{n+1}) = t + 1 > 1$. Then, $t_j(\rho_0 \dots \rho_n) = t$ and we can apply the induction hypothesis and obtain $(\{p_j\}, t) \in \mathrm{Open}_j(\rho_0 \dots \rho_n)$. Furthermore, since the waiting time is not reset to zero at ρ_{n+1}, it holds $p_j \notin l_G(\rho_{n+1})$. Thus, $\mathrm{Emb}_j(\{p_j\}, \rho_{n+1}) = \{p_j\}$ which implies $(\{p_j\}, t + 1) \in \mathrm{Open}_j(\rho_0 \dots \rho_n \rho_{n+1})$.

(ii) We have

$$v_R(\sigma) = \sup_{\tau \in \Gamma_1} v_R(\rho(s_0, \sigma, \tau)) \leq \sup_{\tau \in \Gamma_1} v_P(\rho(s_0, \sigma, \tau)) = v_P(\sigma),$$

where the inequality is due to (i). □

Chapter 5

Solitary PLTL Games

Temporal Logics reason about propositions that change over the course of time. Consider an atomic proposition p. The formula $\mathbf{F}p$ is satisfied, if p is true some time in the future. The formula $\mathbf{G}p$ is satisfied, if p is continuously true from now on. If the time is assumed to be discrete, i.e., the domain of the time steps is discrete, then $\mathbf{X}p$ is satisfied, if p holds after the next time step. If the time is continuous, then $\mathbf{X}$ is disregarded. Derived from Tense Logic [46], Temporal Logic was at first applied to program verification by Pnueli [44]. There, the time domain is given by the executions of a system under consideration, thus it is discrete (typically $\mathbb{N}$). Two paradigms of Temporal Logic can be identified by their treatment of time. The semantics of Linear Temporal Logic [44], (see Section 2.3), are defined with respect to a single time line. In the context of verification, this means that an LTL formula is evaluated with respect to a single execution. The system satisfies the formula, if every execution is a model of the formula. Branching Time Logics like CTL [9] and CTL* [16] on the other hand, are defined with respect to several time lines. They allow to express requirements like *if a request is encountered, then there is the possibility that this request is responded.* However, there might be another execution that does not respond this request. Thus, branching time can be utilized to model unpredictability of the system. The differences between the two paradigms are discussed in [36]. Both flavors of Temporal Logic are today widely used in Formal Verification of closed systems by Model-Checking Tools. We focus on LTL, as we employ formulae as winning conditions to determine the winner of a play.

Typical properties expressible in LTL are liveness properties $\mathbf{F}p$, specifying something good will happen, and safety properties $\mathbf{G}p$, specifying that nothing bad ever happens. Nesting the operators, one can express the Büchi condition, i.e., *infinitely often* p by $\mathbf{GF}p$, for example. Also, Request-Response, Parity and Muller winning conditions are expressible in LTL.

The main advantages of LTL are the variable-free syntax and intuitive semantics, which make it suitable for practical applications. However, there is also a drawback. The semantics of $\mathbf{F}p$ do only require p to be satisfied some time in the future, without specifying a bound. This behavior is undesirable in applications, where the user is generally interested in prompt satisfaction. Also, the semantics of $\mathbf{G}p$, which require p to be true from now on ad infinitum might be to strong, while it would suffice that p holds as long as possible. It is easy to see that there are no such bounds intrinsic to the semantics of LTL (unlike for CTL, where the number of states of the graph provides such a bound [9]), i.e., if $\mathbf{F}p$ is satisfied, then there is also a bound B such that p is true within the next B steps. Consider the formula $\varphi = \mathbf{FG}p$ and the graph G in Figure 5.1. Every path of G is a model of φ, but there is no bound b such that $\mathbf{G}p$ is satisfied within the next b steps on all paths.

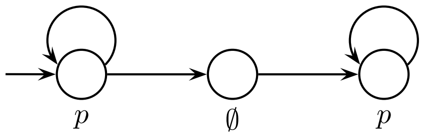

Figure 5.1: A graph without a fixed bound for $\mathbf{FG}p$

One way to overcome this is to subscript the temporal operators by bounds and a direction, i.e., $\mathbf{F}_{\leq 5}$ and $\mathbf{F}_{>5}$ for example. Several so called *real-time* or *quantitative* logics have been introduced with this capability. For example, Metric Temporal Logic [2] or Real-Time CTL [19]. However, they require the user to know the bounds in advance, which is typically not feasible. Also, if a system satisfies a parameterized formula, there is no information on how tight the bounds are. Even more importantly, adding parameterized operators does not increase the expressivity of LTL, as $\mathbf{F}_{\leq 2}p$ is equivalent to the formula $p \vee \mathbf{X}p \vee \mathbf{XX}p$.

To relieve the user from determining the bounds and to allow the search for optimal bounds, the constants can be replaced by variables. Then, $\mathbf{F}_{\leq x}p$ is true, if p is true within the next x steps, where x is a free variable. Satisfaction is then defined with respect to a variable valuation. Thus, given a specification and a system, one can ask whether there is a valuation such that the specification holds, or even determine an optimal valuation. This extension of LTL, called Parametric Linear Temporal Logic, was introduced by Alur et. al. [1].

In this chapter, we use Parametric Linear Temporal Logic to specify winning conditions for infinite games. This is an extension of LTL Games with parameterized winning conditions that allows a clear specification of time-optimal strategies. Player 0's goal is to minimize the waiting time x for formulae $\mathbf{F}_{\leq x}\psi$ and $\mathbf{G}_{>x}\psi$ and to maximize the bound y for formulae $\mathbf{G}_{\leq y}\psi$ and $\mathbf{F}_{>y}$. A PLTL Game $\mathcal{G}$ consists of an arena and a winning condition φ. The winner of a game is determined with respect to a valuation α. Player 0 wins a play of $\mathcal{G}$, if the play is a model of φ with respect to α. Accordingly, Player 0 wins $\mathcal{G}$ with respect to α, if she has a strategy such that every play consistent with this strategy is won by her. This induces a set $\mathcal{W}^0_{\mathcal{G}}$ consisting of the *good* valuations for Player 0. Thus, the questions of emptiness, finiteness and universality arise. Also, one can search for optimal strategies for Player 0.

This chapter is structured as follows. We introduce PLTL in Section 5.1 and present some simple results from [1] which encompass the typical treatment of a new logic. After defining syntax and semantics, we consider dualities of the temporal operators and show that the set of parameterized temporal operators can be restricted to $\mathbf{F}_{\leq x}$ and $\mathbf{G}_{\leq y}$ without losing any expressive power. The formulae built from the classical temporal operators and $\mathbf{F}_{\leq x}$ form the fragment $\mathrm{PLTL}_{\mathbf{F}}$ while the fragment $\mathrm{PLTL}_{\mathbf{G}}$ is defined analogously. Most of the remainder of this chapter is concerned with these unipolar fragments, which have good decidability properties while the questions for the full logic are mostly open [1]. However, there is evidence that the full logic is too strong. We reiterate the findings at the end of Section 5.1. As Model-Checking is a special case of synthesis, we restrict our attention to these fragments when defining PLTL Games in Section 5.2. The main part of this section is devoted to two technical lemmata that are the key to the main theorems about PLTL Games. For unipolar solitary games, it is decidable whether Player i wins a PLTL Game with respect to some, all or only finitely many valuations. Furthermore, for these games optimal valuations are computable. For solitary $\mathrm{PLTL}_{\mathbf{G}}$ Games, a proof from [1] can be adapted to our cause easily and another theorem (without proof) from that paper can be adapted to solitary $\mathrm{PLTL}_{\mathbf{F}}$ Games. Our technical lemma for $\mathrm{PLTL}_{\mathbf{F}}$ games establishes a similar result (with slightly higher bounds). We close this chapter by considering alternative semantics for PLTL Games in Section 5.3. Instead of requiring that the winning condition holds on all plays with respect to a fixed valuation, it requires that there is a valuation for every play, such that the winning condition holds with respect to that valuation on that particular play. Thus, the valuation is not uniform for all plays, but there might be different valuations for different plays. We analyze these semantics and draw comparisons to the standard semantics introduced in Section 5.2.

5.1 Parametric Linear Temporal Logic

Parametric Linear Temporal Logic extends Linear Temporal Logic with temporal operators that can be subscripted by variables. We begin this section by defining the syntax and semantics of PLTL and by discussing some elementary properties before we close it by stating the results.

Let P be a set of atomic propositions and $\mathcal{Y}$ and $\mathcal{X}$ be two disjoint sets of *variables*. Parametric Linear Temporal Logic, PLTL for short, extends the syntax of Linear Temporal Logic introduced in Section 2.3 by a parameterized version of every temporal operator (but $\mathbf{X}$, which has no reasonable parameterized version). A *parameter* is either a variable or a non-negative integer, a *constant* parameter. We explain the need for two disjoint sets of variables later in this section. We define the set PLTL of *Parametric Linear Temporal Logic* formulae inductively by

- $p, \neg p \in$ PLTL if $p \in P$,
- $\varphi \wedge \psi, \varphi \vee \psi \in$ PLTL if $\varphi, \psi \in$ PLTL,

- $\mathbf{X}\varphi, \mathbf{F}\varphi, \mathbf{G}\varphi, \varphi\mathbf{U}\psi, \varphi\mathbf{R}\psi \in \text{PLTL}$ if $\varphi, \psi \in \text{PLTL}$,
- $\mathbf{F}_{\leq x}\varphi, \mathbf{G}_{\leq y}\varphi, \varphi\mathbf{U}_{\leq x}\psi, \varphi\mathbf{R}_{\leq y}\psi \in \text{PLTL}$ if $\varphi, \psi \in \text{PLTL}$, $x \in \mathcal{X} \cup \mathbb{N}$, $y \in \mathcal{Y} \cup \mathbb{N}$, and
- $\mathbf{F}_{>y}\varphi, \mathbf{G}_{>x}\varphi, \varphi\mathbf{U}_{>y}\psi, \varphi\mathbf{R}_{>x}\psi \in \text{PLTL}$ if $\varphi, \psi \in \text{PLTL}$, $x \in \mathcal{X} \cup \mathbb{N}$, $y \in \mathcal{Y} \cup \mathbb{N}$.

There are several complexity measures of a formula φ that are used later on. The *set of variables of* φ, $\text{var}(\varphi)$, and the *set of constants of* φ, $\text{con}(\varphi)$, are defined in the obvious way. Furthermore, let

- n_φ be the number of distinct subformulae of φ,
- k_φ be the number of temporal operators of φ parameterized by a variable, and
- c_φ be the product of all non-zero constants that parameterize operators in φ.

To define the semantics of PLTL, we need to know the values of the variables: a *valuation* is a mapping $\alpha : \mathcal{X} \cup \mathcal{Y} \to \mathbb{N}$. By convention, we compare valuations componentwise, i.e., $\alpha \leq \beta$ iff $\alpha(z) \leq \beta(z)$ for all $z \in \mathcal{X} \cup \mathcal{Y}$. Thus, $\leq$ is a partial order on the set of all valuations, and we can speak of upwards-closed and downwards-closed sets of valuations. Oftentimes, when dealing with a fixed formula φ, we implicitly assume that a valuation is a mapping $\alpha : \text{var}(\varphi) \to \mathbb{N}$, if it is opportune. The following facts are useful throughout this chapter.

Remark 5.1. *Let α_0 be the valuation that maps every variable to zero.*

(i) A downwards-closed set of valuations is non-empty iff it contains α_0.

(ii) An upwards-closed set of valuations is universal iff it contains α_0.

(iii) An upwards-closed set of valuations is infinite iff it is non-empty.

To deal with temporal operators parameterized by constants more conveniently, we extend the domain of α to $\mathcal{X} \cup \mathcal{Y} \cup \mathbb{N}$ and define $\alpha(c) = c$ for all $c \in \mathbb{N}$. Let $\rho = \rho_0\rho_1\rho_2\ldots$ be a path of a labeled graph (V, E, l). *Satisfaction of a formula φ instantiated by α at position n of ρ*, written $(\rho, n, \alpha) \models \varphi$, is defined inductively by

- $(\rho, n, \alpha) \models p$ iff $p \in l(\rho_n)$,
- $(\rho, n, \alpha) \models \neg p$ iff $p \notin l(\rho_n)$,
- $(\rho, n, \alpha) \models \varphi \wedge \psi$ iff $(\rho, n, \alpha) \models \varphi$ and $(\rho, n, \alpha) \models \psi$
- $(\rho, n, \alpha) \models \varphi \vee \psi$ iff $(\rho, n, \alpha) \models \varphi$ or $(\rho, n, \alpha) \models \psi$,
- $(\rho, n, \alpha) \models \mathbf{X}\varphi$ iff $(\rho, n+1, \alpha) \models \varphi$,
- $(\rho, n, \alpha) \models \mathbf{F}\varphi$ iff there exists $k \geq 0$ such that $(\rho, n+k, \alpha) \models \varphi$,
- $(\rho, n, \alpha) \models \mathbf{G}\varphi$ iff for all ≥ 0: $(\rho, n+k, \alpha) \models \varphi$,
- $(\rho, n, \alpha) \models \varphi\mathbf{U}\psi$ iff there exists $k \geq 0$ such that $(\rho, n+k, \alpha) \models \psi$ and for all l such that $0 \leq l < k$: $(\rho, n+l, \alpha) \models \psi$,
- $(\rho, n, \alpha) \models \varphi\mathbf{R}\psi$ iff for all $k \geq 0$: either $(\rho, n+k, \alpha) \models \psi$ or there exists $l < k$ such that $(\rho, n+l, \alpha) \models \varphi$,

- $(\rho, n, \alpha) \models \mathbf{F}_{\leq x}\varphi$ iff there exists $k \leq \alpha(x)$ such that $(\rho, n+k, \alpha) \models \varphi$,
- $(\rho, n, \alpha) \models \mathbf{G}_{\leq y}\varphi$ iff for all $k \leq \alpha(y)$: $(\rho, n+k, \alpha) \models \varphi$,
- $(\rho, n, \alpha) \models \varphi\mathbf{U}_{\leq x}\psi$ iff there exists $k \leq \alpha(x)$ such that $(\rho, n+k, \alpha) \models \psi$ and for all l such that $0 \leq l < k$: $(\rho, n+l, \alpha) \models \psi$,
- $(\rho, n, \alpha) \models \varphi\mathbf{R}_{\leq y}\psi$ iff for all $k \leq \alpha(y)$: either $(\rho, n+k, \alpha) \models \psi$ or there exists $l < k$ such that $(\rho, n+l, \alpha) \models \varphi$,
- $(\rho, n, \alpha) \models \mathbf{F}_{>y}\varphi$ iff there exists $k > \alpha(y)$ such that $(\rho, n+k, \alpha) \models \varphi$,
- $(\rho, n, \alpha) \models \mathbf{G}_{>x}\varphi$ iff for all $k > \alpha(x)$: $(\rho, n+k, \alpha) \models \varphi$,
- $(\rho, n, \alpha) \models \varphi\mathbf{U}_{>y}\psi$ iff there exists $k > \alpha(y)$ such that $(\rho, n+k, \alpha) \models \psi$ and for all l such that $0 \leq l < k$: $(\rho, n+l, \alpha) \models \psi$, and
- $(\rho, n, \alpha) \models \varphi\mathbf{R}_{>x}\psi$ iff for all $k > \alpha(x)$: either $(\rho, n+k, \alpha) \models \psi$ or there exists $l < k$ such that $(\rho, n+l, \alpha) \models \varphi$.

Finally, ρ is a model of φ instantiated by α, $(\rho, \alpha) \models \varphi$ iff $(\rho, 0, \alpha) \models \varphi$.

A valuation α makes φ *satisfiable*, if there exists a labeled path ρ of some labeled graph such that $(\rho, \alpha) \models \varphi$. Analogously, α makes φ *valid*, if $(\rho, \alpha) \models \varphi$ for all paths ρ of all labeled graphs. This defines the sets $S(\varphi)$ and $V(\varphi)$ containing the valuations that make φ satisfiable respectively valid.

Let G be a fixed labeled graph. We say that α makes φ *satisfiable in* G if there exists a path ρ of G such that $(\rho, 0, \alpha) \models \varphi$, and that α makes φ *valid in* G if $(\rho, 0, \alpha) \models \varphi$ for every path ρ of G. This defines the sets $S(G, \varphi)$ and $V(G, \varphi)$ containing the valuations that make φ satisfiable respectively valid in G. Problems regarding $S(\varphi)$ can obviously be reduced to the same problems for $S(G, \varphi)$, where G is a graph whose paths coincide with all labeled paths for the propositions occurring in φ. The same observation can be made for the validity sets $V(\varphi)$ and $V(G, \varphi)$.

Remark 5.2. *Although, we do only allow negation of atomic propositions,* PLTL *is closed under negation as witnessed by the dualities for* LTL *and the following dualities of the parameterized operators.*

$$\begin{aligned}
\neg\mathbf{F}_{\leq x}\varphi &\equiv \mathbf{G}_{\leq y}\neg\varphi \\
\neg\mathbf{F}_{>x}\varphi &\equiv \mathbf{G}_{>y}\neg\varphi \\
\neg(\varphi\mathbf{U}_{\leq x}\psi) &\equiv (\neg\varphi)\mathbf{R}_{\leq y}(\neg\psi) \\
\neg(\varphi\mathbf{U}_{>x}\psi) &\equiv (\neg\varphi)\mathbf{R}_{>y}(\neg\psi)
\end{aligned}$$

Thus, every negation can be pushed to the atomic formulae p and $\neg p$. Removing double negations then gives an equivalent formula in negation normal form, as it is required by the syntax of PLTL.

We abuse our notation slightly and write $\neg\varphi$ for the PLTL formula obtained by this procedure.

Remark 5.3. *Let* φ *be a* PLTL *formula and* G *a labeled graph. Then,*

(i) $S(\varphi) = V(\neg\varphi)$, *and*

(ii) $S(G, \varphi) = V(G, \neg\varphi)$.

As with classical LTL, many temporal operators are just syntactic sugar, as they can be defined in terms of a small set of basic operators. We state some straightforward equivalences and dualities that allow us to supersede all but two parameterized operators [1].

$$\begin{aligned}
\mathbf{F}\varphi &\equiv \mathbf{tt}\mathbf{U}\varphi \\
\mathbf{G}\varphi &\equiv \mathbf{ff}\mathbf{R}\varphi \\
\varphi\mathbf{U}_{\leq x}\psi &\equiv (\varphi\mathbf{U}\psi) \wedge \mathbf{F}_{\leq x}\psi \\
\varphi\mathbf{R}_{\leq y}\psi &\equiv (\varphi\mathbf{R}\psi) \vee \mathbf{G}_{\leq y}\psi \\
\mathbf{F}_{>y}\varphi &\equiv \mathbf{G}_{\leq y}\mathbf{F}\mathbf{X}\varphi \\
\mathbf{G}_{>x}\varphi &\equiv \mathbf{F}_{\leq x}\mathbf{G}\mathbf{X}\varphi \\
\varphi\mathbf{U}_{>y}\psi &\equiv \mathbf{G}_{\leq y}(\varphi \wedge \mathbf{X}(\varphi\mathbf{U}\psi)) \\
\varphi\mathbf{R}_{>x}\psi &\equiv \mathbf{F}_{\leq x}(\varphi \vee \mathbf{X}(\varphi\mathbf{R}\psi))
\end{aligned}$$

This allows us to restrict our attention to the parameterized operators $\mathbf{F}_{\leq x}$ and $\mathbf{G}_{\leq y}$ along with the unparameterized operators $\mathbf{X}$, $\mathbf{U}$, and $\mathbf{R}$ while still retaining closure under negation, since $\mathbf{F}_{\leq x}$ and $\mathbf{G}_{\leq y}$ are dual. Also, replacing the operators leads only to a linear increase of the formula's complexity. The number of distinct subformulae n_φ grows by a constant number in every rewriting step. The number of subformulae parameterized with variables k_φ and the product of the non-zero constants c_φ remain unchanged. Also, the variables used in the original formula and in the rewritten formula are the same. The length measured in symbols might grow exponentially, though.

The parameterized operators are obviously monotone, either upwards as $\mathbf{F}_{\leq y}$ or downwards as $\mathbf{G}_{\leq y}$, for example. Consider a parameterized eventuality: if it is satisfied in no more than k steps, then it also satisfied in no more than $k+1$ steps. Dually, consider a parameterized always: if a formula holds for the next k steps, then it also holds for the next $k-1$ steps. The same observations can be made for all other parameterized operators, also.

Remark 5.4 ([1])**.** *Let* α, β *be valuations such that* $\alpha(x) < \beta(x)$ *and* $\alpha(y) > \beta(y)$ *for some* $x \in \mathcal{X} \cup \mathbb{N}$ *and* $y \in \mathcal{Y} \cup \mathbb{N}$. *Then,*

$$\begin{aligned}
(\rho, n, \alpha) \models \mathbf{F}_{\leq x}\varphi \quad &\textit{implies} \quad (\rho, n, \beta) \models \mathbf{F}_{\leq x}\varphi, \\
(\rho, n, \alpha) \models \mathbf{G}_{\leq y}\varphi \quad &\textit{implies} \quad (\rho, n, \beta) \models \mathbf{G}_{\leq y}\varphi, \\
(\rho, n, \alpha) \models \varphi\mathbf{U}_{\leq x}\psi \quad &\textit{implies} \quad (\rho, n, \beta) \models \varphi\mathbf{U}_{\leq x}\psi, \\
(\rho, n, \alpha) \models \varphi\mathbf{R}_{\leq y}\psi \quad &\textit{implies} \quad (\rho, n, \beta) \models \varphi\mathbf{R}_{\leq y}\psi,
\end{aligned}$$

$$\begin{aligned}
(\rho, n, \alpha) &\models \mathbf{F}_{>y}\varphi \quad \textit{implies} \quad (\rho, n, \beta) \models \mathbf{F}_{>y}\varphi, \\
(\rho, n, \alpha) &\models \mathbf{G}_{>x}\varphi \quad \textit{implies} \quad (\rho, n, \beta) \models \mathbf{G}_{>x}\varphi, \\
(\rho, n, \alpha) &\models \varphi\mathbf{U}_{>y}\psi \quad \textit{implies} \quad (\rho, n, \beta) \models \varphi\mathbf{U}_{>y}\psi, \textit{ and} \\
(\rho, n, \alpha) &\models \varphi\mathbf{R}_{>x}\psi \quad \textit{implies} \quad (\rho, n, \beta) \models \varphi\mathbf{R}_{>x}\psi.
\end{aligned}$$

Therefore, the operators parameterized with $x \in \mathcal{X}$ are called *upwards-monotone*, while the operators parameterized with $y \in \mathcal{Y}$ are called *downwards-monotone*. The sets $S(\varphi)$, $V(\varphi)$ and their analogons for labeled graphs are upwards-closed respectively downwards-closed, if φ contains only upwards-monotone respectively downwards-monotone operators. This can be shown by an easy induction over the construction of φ, applying Remark 5.4 for every parameterized operator. Since we can express all upwards-monotone operators with $\mathbf{F}_{\leq x}$ and all downwards-monotone operators with $\mathbf{G}_{\leq y}$, these fragments can be defined by allowing only one of the two parameterized operators: the fragment $\text{PLTL}_\mathbf{F}$ consists of all formulae build from atomic propositions and their negations by the boolean connectives, the standard temporal operators, temporal operators parameterized by constants $c \in \mathbb{N}$ and $\mathbf{F}_{\leq x}$ for $x \in \mathcal{X}$. Analogously, the fragment $\text{PLTL}_\mathbf{G}$ consists of all formulae build from the atomic propositions and their negations by the boolean connectives, the standard temporal operators, temporal operators parameterized by constants $c \in \mathbb{N}$ and $\mathbf{G}_{\leq y}$ for $y \in \mathcal{Y}$. Since the monotonicity of a parameterized operator can be seen as its polarity, we call formulae in those fragments *unipolar*. As $\mathbf{F}_{\leq x}$ and $\mathbf{G}_{\leq y}$ are dual, the fragments are dual as well.

Remark 5.5. *If* $\varphi \in \text{PLTL}_\mathbf{F}$*, then* $\neg\varphi \in \text{PLTL}_\mathbf{G}$ *and if* $\varphi \in \text{PLTL}_\mathbf{G}$*, then* $\neg\varphi \in \text{PLTL}_\mathbf{F}$*.*

For a formula φ and a valuation α, let $\alpha(\varphi)$ be the formula obtained by replacing every variable $z \in \mathcal{X} \cup \mathcal{Y}$ by $\alpha(z)$. The resulting formula is a PLTL formula without variables, i.e., every parameterized operator is parameterized by a constant.

Lemma 5.6 ([1])**.** *For every valuation* α *and every* PLTL *formula* φ*, there is an* LTL *formula* φ' *such that for all* ρ *and all* n

$$(\rho, n, \alpha) \models \varphi \Leftrightarrow (\rho, n, \alpha) \models \alpha(\varphi) \Leftrightarrow (\rho, n) \models \varphi'.$$

Proof. The first equivalence is trivial. For the second, define $\psi_\wedge^n$ and $\psi_\vee^n$ by $\psi_\wedge^0 = \psi_\vee^0 = \psi$ and $\psi_\wedge^{j+1} = \psi \wedge \mathbf{X}\psi_\wedge^j$ respectively $\psi_\vee^{j+1} = \psi \vee \mathbf{X}\psi_\vee^j$. Then, the following equivalences hold.

$$\begin{aligned}
(\rho, n, \alpha) &\models \mathbf{F}_{\leq k}\psi \quad \text{iff} \quad (\rho, n) \models \psi_\vee^k \\
(\rho, n, \alpha) &\models \mathbf{G}_{\leq k}\psi \quad \text{iff} \quad (\rho, n) \models \psi_\wedge^k
\end{aligned}$$

Thus, we can inductively replace every subformula $\mathbf{F}_{\leq k}\psi$ or $\mathbf{G}_{\leq k}\psi$ of $\alpha(\varphi)$ by an LTL formula. This suffices by the above remarks. ☐

In the following we use $\alpha(\varphi)$ also to denote the LTL formula φ' equivalent to the PLTL formula $\alpha(\varphi)$, if it is clear from the context which formula we mean. The size of the LTL formula $\alpha(\varphi)$ is linear in $n_\varphi + \sum_{z \in \mathrm{var}(\varphi)} \alpha(z) + \sum_{c \in \mathrm{con}(\varphi)} c$, which is exponential in the size of α, if we use a binary encoding for $\alpha(z)$. Lemma 5.6 allows us to reduce satisfiability and validity of PLTL formulae with respect to a fixed valuation α to the corresponding problems for LTL formulae.

We close this section by presenting the results of [1].

Theorem 5.7 ([1]). *Let φ be a unipolar* PLTL *formula and G a labeled graph. The emptiness, universality, and finiteness problem for $S(\varphi)$ and $S(G, \varphi)$ are decidable.*

The results for $\mathrm{PLTL}_{\mathbf{G}}$ formulae are proven, and for $\mathrm{PLTL}_{\mathbf{F}}$ formulae a proof idea is presented. We will adapt these proofs to games in Section 5.2. From these results, one can also easily derive the solution to several natural optimization problems.

Theorem 5.8 ([1]). *Let G a labeled graph.*

(i) Let φ be a $\mathrm{PLTL}_{\mathbf{G}}$ *formula.*

- $\max_{\alpha \in S(\varphi)} \max_{y \in \mathrm{var}(\varphi)} \alpha(y)$,
- $\max_{\alpha \in S(\varphi)} \min y \in \mathrm{var}(\varphi) \alpha(y)$,
- $\max_{\alpha \in S(G,\varphi)} \max_{y \in \mathrm{var}(\varphi)} \alpha(y)$, *and*
- $\max_{\alpha \in S(G,\varphi)} \min y \in \mathrm{var}(\varphi) \alpha(y)$

are computable.

(ii) Let φ be a $\mathrm{PLTL}_{\mathbf{F}}$ *formula.*

- $\min_{\alpha \in S(\varphi)} \min_{y \in \mathrm{var}(\varphi)} \alpha(y)$,
- $\min_{\alpha \in S(\varphi)} \max y \in \mathrm{var}(\varphi) \alpha(y)$,
- $\min_{\alpha \in S(G,\varphi)} \min_{y \in \mathrm{var}(\varphi)} \alpha(y)$, *and*
- $\min_{\alpha \in S(G,\varphi)} \max y \in \mathrm{var}(\varphi) \alpha(y)$

are computable.

The situation for full PLTL is more complicated. While emptiness and universality for $S(\varphi)$ are still decidable, the question whether $S(\varphi)$ contains a valuation α such that $\alpha(x) = \alpha(y)$ is undecidable. This can be seen easily by considering an even stronger logic. If we allow temporal operators subscripted with $= x$ (with the obvious semantics), then the logic can encode terminating runs of a Turing Machine. Then, all interesting questions, like the emptiness of $S(\varphi)$, become undecidable. Also, this explains the need for two disjoint sets of variables, one for the downwards-monotone operators and one for the upwards-monotone operators. If we allow z to parameterize both types of operators, then equality subscripts can be encoded easily.

5.2 PLTL Games

In this section, we will define games with winning conditions in PLTL, focusing on unipolar formulae, again. After discussing some simple observations about these games, we will prove two technical lemmata, on which the main theorems of this section rely.

A (initialized) PLTL Game $\mathcal{G} = (G, s, \varphi)$ consists of an arena G, an initial vertex s of the arena and a winning condition $\varphi \in$ PLTL. Additionally, we say that $\mathcal{G}$ is a $\text{PLTL}_{\mathbf{F}}$ *Game* if $\varphi \in \text{PLTL}_{\mathbf{F}}$, and similarly, that $\mathcal{G}$ is a $\text{PLTL}_{\mathbf{G}}$ *Game* if $\varphi \in \text{PLTL}_{\mathbf{G}}$. We call these games *unipolar*.

The definition of a PLTL Game does not fit properly into the framework for games we defined in Section 2.4, since φ alone does not define a set of winning plays. Paired with a valuation α, we define $\rho \in \text{Win} \Leftrightarrow (\rho, \alpha) \models \varphi$. We say Player i *wins* $\mathcal{G}$ *with respect to* α if she has a winning strategy with respect to α, i.e., every play consistent with σ is a model of the winning condition with respect to α. Since φ for a fixed α is equivalent to an LTL formula, a game for a fixed α is nothing more than an LTL game (albeit the PLTL formula is presumably shorter than the equivalent LTL formula). On the positive side, this means that a PLTL Game for a fixed α can be solved using the techniques for LTL Games and finite-state determinacy carries over. On the negative side, this means that a PLTL game for a fixed valuation is not particularly interesting. Therefore, we are not interested in a single valuation but in the set of valuations such that Player 0 has a winning strategy for the game with these valuations. Since we deal with a fixed game, we can assume that valuations are only defined for the variables appearing in φ. Formally, we define the set of valuations that let Player i win $\mathcal{G}$,

$$\mathcal{W}_{\mathcal{G}}^{i} = \{\alpha \mid \text{Player } i \text{ wins } (G, s, \alpha(\varphi))\}.$$

As we have seen above, the membership problem $\alpha \in \mathcal{W}_{\mathcal{G}}^{i}$ can be solved by determining the winner of the LTL game $(G, s, \alpha(\varphi))$. So, we turn our attention to the questions of emptiness, universality, and finiteness of the set $\mathcal{W}_{\mathcal{G}}^{i}$. Since games are tightly related to satisfiability and validity of their winning conditions, we also focus on the unipolar fragments, which turned out to have nice decidability properties. Also, the strategy problem is no longer a decision problem, but more of an optimization problem: what is the *optimal* valuation α, such that Player i can win $\mathcal{G}$ with respect to α. The optimality of a valuation depends on the winning condition φ. A natural approach is to minimize $\alpha(x)$ for the variables x of parameterized eventualities $\mathbf{F}_{\leq x}$, and to maximize $\alpha(y)$ for the variables y of parameterized always' $\mathbf{G}_{\leq y}$. Since there might be a trade-off between maximizing some values and minimizing the others, we again retreat to unipolar games, as there is a natural preference order for valuations in these cases.

Example 5.9. We have seen in Chapter 3 that the Request-Response winning condition can be expressed by an LTL formulae. The eventualities can be parameterized by variables to find optimal global bounds on the waiting times. Given a Request-Response Game $\mathcal{G} = (G, s, (Q_j, P_j)_{j=1,\dots,k})$ we label G by $l(s) = \{q_j \mid s \in Q_j\} \cup \{p_j \mid s \in P_j\}$ and define the $\mathrm{PLTL}_{\mathbf{F}}$ Game $\mathcal{G}' = (G, s, \varphi)$ where

$$\varphi := \bigwedge_{j=1}^{k} \mathbf{G}\left(q_j \rightarrow \mathbf{F}_{\leq x_j} p_j\right).$$

Alternatively, the eventualities can be parameterized by a single variable, thereby enforcing the same bound for all conditions.

Lemma 5.10. *Let $\mathcal{G}$ be a Request-Response Game and $\mathcal{G}'$ the corresponding* $\mathrm{PLTL}_{\mathbf{F}}$ *Game. Player* 0 *wins $\mathcal{G}$ iff $\mathcal{W}^0_{\mathcal{G}'} \neq \emptyset$. Furthermore, if $\alpha \in \mathcal{W}^0_{\mathcal{G}'}$, then there exists a winning strategy σ for Player* 0 *that bounds the waiting time for condition j to $\alpha(x_j)$.*

Proof. If Player 0 wins $\mathcal{G}$, then she also has a finite-state winning strategy σ of size $k2^{k+1}$, by Theorem 3.1, that bounds the waiting times to $|G| \cdot k$ by Corollary 3.6. Let $\alpha(x_j) = |G| \cdot k$ for all x_j. Then, σ is a winning strategy for $\mathcal{G}$ with respect to α. Thus, $\alpha \in \mathcal{W}^0_{\mathcal{G}}$. The other direction and the second statement are obvious. □

In an LTL Game with winning condition φ, Player 0's goal is to move the token such that every play is a model of φ. If she cannot fulfill this, i.e., she has no winning strategy, then Player 1 has a winning strategy for this game. So, he can move the token in a way that every play is a model of $\neg\varphi$. If we swap the roles of the two players, then Player 0 has a winning strategy for the game with winning condition $\neg\varphi$. This duality is useful throughout this chapter. Formally, for an arena $G = (V, V_0, V_1, E)$, let $\overline{G} = (V, V_1, V_0, E)$ be the *dual arena*, where the two players swap their positions. Obviously, the dual arena of $\overline{G}$ is G. Also, a strategy for one of the players in G is a strategy for the other player in $\overline{G}$. Given a PLTL Game $\mathcal{G} = (G, s_0, \varphi)$, the *dual game* is $\overline{\mathcal{G}} = (\overline{G}, s_0, \neg\varphi)$. Remark 5.5 implies that the dual game of a $\mathrm{PLTL}_{\mathbf{G}}$ Game is a $\mathrm{PLTL}_{\mathbf{F}}$ Game and vice versa.

Lemma 5.11. *Let α be a valuation and $\mathcal{G}$ a* PLTL *Game. Player i wins $\mathcal{G}$ with respect to α iff Player $1 - i$ wins $\overline{\mathcal{G}}$ with respect to α.*

Proof. Player 0 wins $(G, s_0, \alpha(\varphi))$ iff she has a strategy σ such that every play consistent with σ is a model of $\alpha(\varphi)$. Thus, the same strategy σ is a strategy for Player 1 in $\overline{G}$ such that all plays consistent with σ are models of $\alpha(\varphi)$. Hence, Player 1 wins $(\overline{G}, s_0, \neg\varphi)$. The reasoning for Player 1 is analogous due to determinacy of LTL Games. For the other direction of the statement, use $\neg\neg\varphi = \varphi$ and $\overline{\overline{G}} = G$. □

The sets $\mathcal{W}^i_{\mathcal{G}}$ enjoy two types of dualities, which we rely on in the following. The first one is due to determinacy of LTL Games, by Theorem 2.18.

Lemma 5.12. *Let $\mathcal{G}$ be a* PLTL *Game. Then*

(i) $\mathcal{W}^0_{\mathcal{G}}$ is the complement of $\mathcal{W}^1_{\mathcal{G}}$.
(ii) $\mathcal{W}^i_{\mathcal{G}} = \mathcal{W}^{1-i}_{\overline{\mathcal{G}}}$.

The monotonicity of the parameterized operators gives rise to the first results.

Lemma 5.13. *Let $\mathcal{G}_\mathbf{G}$ be a* PLTL$_\mathbf{G}$ *Game and $\mathcal{G}_\mathbf{F}$ be a* PLTL$_\mathbf{F}$ *Game.*

(i) The sets $\mathcal{W}^0_{\mathcal{G}_\mathbf{G}}$ and $\mathcal{W}^1_{\mathcal{G}_\mathbf{F}}$ are downwards-closed.

(ii) The sets $\mathcal{W}^0_{\mathcal{G}_\mathbf{F}}$ and $\mathcal{W}^1_{\mathcal{G}_\mathbf{G}}$ are upwards-closed.

Proof. We do the proofs for $\mathcal{W}^0_{\mathcal{G}_\mathbf{G}}$ and $\mathcal{W}^0_{\mathcal{G}_\mathbf{F}}$. For the other sets, apply Lemma 5.12 (ii) and Remark 5.5. Let $\varphi_\mathbf{G}$ and $\varphi_\mathbf{F}$ be the winning conditions of $\mathcal{G}_\mathbf{G}$ respectively $\mathcal{G}_\mathbf{F}$.

(i) Let $\alpha \in \mathcal{W}^0_{\mathcal{G}_\mathbf{G}}$ and $\alpha \geq \beta$. Then, Player 0 has a winning strategy σ for $\mathcal{G}_\mathbf{G}$ with respect to α, i.e., $(\rho, \alpha) \models \varphi_\mathbf{G}$ for every play ρ consistent with σ. Then, an easy induction over the structure of φ_G with repeated applications of Remark 5.4 for every parameterized always yields $(\rho, \beta) \models \varphi_\mathbf{G}$. Thus, σ is also a winning strategy for $\mathcal{G}_\mathbf{G}$ with respect to β.

(ii) The proof goes along the lines of (i): if Player 0 has a winning strategy σ for $\mathcal{G}_\mathbf{F}$ with respect to α, then every play consistent with σ is a model of the winning condition with respect to α. Applying Remark 5.4, one can easily show that every such play is also a model with respect to β, which finishes the proof. □

Combining these closure properties with Remark 5.1 yields the first decision procedures.

Corollary 5.14. *Let $\mathcal{G}_\mathbf{G}$ be a* PLTL$_\mathbf{G}$ *Game and $\mathcal{G}_\mathbf{F}$ be a* PLTL$_\mathbf{F}$ *Game, and α_0 the valuation that maps every variable to zero.*

(i) The emptiness of $\mathcal{W}^0_{\mathcal{G}_\mathbf{G}}$ and $\mathcal{W}^1_{\mathcal{G}_\mathbf{F}}$ is decidable by determining the winner of $\mathcal{G}_\mathbf{G}$ respectively $\mathcal{G}_\mathbf{F}$ with respect to α_0.

(ii) The universality of $\mathcal{W}^0_{\mathcal{G}_\mathbf{F}}$ and $\mathcal{W}^1_{\mathcal{G}_\mathbf{G}}$ is decidable by determining the winner of $\mathcal{G}_\mathbf{F}$ respectively $\mathcal{G}_\mathbf{G}$ with respect to α_0.

The finiteness problem for a PLTL$_\mathbf{F}$ Game coincides with the emptiness problem by exploiting upwards-closure of $\mathcal{W}^0_{\mathcal{G}_\mathbf{F}}$.

Remark 5.15. *Let $\mathcal{G}_\mathbf{F} = (G, s, \varphi)$ be a* PLTL$_\mathbf{F}$ *Game. Then, $\mathcal{W}^0_{\mathcal{G}_\mathbf{F}}$ is infinite iff it is non-empty.*

The finiteness problem for a PLTL$_\mathbf{G}$ Game $\mathcal{G} = (G, s, \varphi)$ can be reduced to the universality problem for a (simpler) PLTL$_\mathbf{G}$ game. We assume that φ has at least one temporal operator parameterized with a variable, since the problem is trivial, otherwise. The set $\mathcal{W}^0_\mathcal{G}$ is infinite iff there is a variable $y \in \text{var}(\varphi)$ such that y is mapped to infinitely many values by the valuations in $\mathcal{W}^0_\mathcal{G}$. By downwards-closure we can assume that all other variables are mapped to zero. Furthermore, if y is mapped to infinitely many values, then it is mapped to all possible values, again by downwards-closure. To combine this, we define φ_y to be the formula obtained from φ by replacing every variable $z \neq y$ by 0 and $\mathcal{G}_y = (G, s, \varphi_y)$.

Lemma 5.16. *Let* $\mathcal{G} = (G, s, \varphi)$ *be a* $\mathrm{PLTL}_{\mathbf{G}}$ *Game and* $\mathcal{G}_y$ *defined as above.* $\mathcal{W}^0_{\mathcal{G}}$ *is infinite, iff* $\mathcal{W}^0_{\mathcal{G}_y}$ *is universal for some variable* $y \in \mathrm{var}(\varphi)$*.*

Proof. Let $\mathcal{W}^0_{\mathcal{G}}$ be infinite. Then, there is a variable $y \in \mathrm{var}(\varphi)$ such that y is mapped to infinitely many different values by the valuations in $\mathcal{W}^0_{\mathcal{G}}$. Without loss of generality, we can assume that every variable $z \neq y$ is mapped to zero by each one of the infinitely many valuations, due to downwards-closure. Restrictions of these valuations are also contained in $\mathcal{W}^0_{\mathcal{G}_y}$. Thus, $\mathcal{W}^0_{\mathcal{G}_y}$ is infinite as well. Since there is only a single variable in φ_y, $\mathcal{W}^0_{\mathcal{G}} \subseteq \mathbb{N}$. Every infinite, downwards-closed subset of $\mathbb{N}$ is equal to $\mathbb{N}$. Thus, $\mathcal{W}^0_{\mathcal{G}_y}$ is universal.

Conversely, every $\alpha \in \mathcal{W}^0_{\mathcal{G}_y}$ can be expanded to a valuation α' for $\mathcal{G}$ by mapping $z \neq y$ to zero, which is contained in $\mathcal{W}^0_{\mathcal{G}}$. Thus, $\mathcal{W}^0_{\mathcal{G}}$ is infinite, if $\mathcal{W}^0_{\mathcal{G}_y}$ is universal. □

So, to decide whether $\mathcal{W}^0_{\mathcal{G}_{\mathbf{G}}}$ is infinite can be done by solving k_φ many universality problems. Note also that φ_y has only one variable; hence, $k_{\varphi_y} = 1$, which decreases the size of the bound in Corollary 5.22, which proves the decidability of the universality problem for solitary $\mathrm{PLTL}_{\mathbf{G}}$ Games.

Before we turn our attention to solitary games, we state two Lemmata that apply the results obtained for solitary games to the satisfiability and validity problem for PLTL and to non-solitary games.

Remark 5.17. *Let* φ *be a* PLTL *formula and* $G = (V, E, l)$ *a labeled graph. Define the solitary arenas* $G_0 = (G, V, \emptyset, E)$ *and* $G_1 = (G, \emptyset, V, E)$ *and the solitary games* $\mathcal{G}_0 = (G_0, s, \varphi)$ *and* $\mathcal{G}_1 = (G_1, s, \varphi)$*. Then,* $S(G, \varphi) = \mathcal{W}^0_{\mathcal{G}_0}$ *and* $V(G, \varphi) = \mathcal{W}^0_{\mathcal{G}_1}$*.*

So, the work of [1] can be embedded into our game-theoretic framework and our results are applicable.

Lemma 5.18. *Let* $\mathcal{G} = (G, s, \varphi)$ *be a* PLTL *Game with arena* $G = (V, V_0, V_1, E)$*. Define the solitary arenas* $G_0 = (V, V, \emptyset, E)$ *and* $G_1 = (V, \emptyset, V, E)$ *and the solitary games* PLTL $\mathcal{G}_0 = (G_0, s, \varphi)$ *and* $\mathcal{G}_1 = (G_1, s, \varphi)$ *for Player* 0 *respectively* 1*. Then,* $\mathcal{W}^0_{\mathcal{G}_1} \subseteq \mathcal{W}^0_{\mathcal{G}} \subseteq \mathcal{W}^0_{\mathcal{G}_0}$ *and* $\mathcal{W}^1_{\mathcal{G}_1} \supseteq \mathcal{W}^1_{\mathcal{G}} \supseteq \mathcal{W}^1_{\mathcal{G}_0}$*.*

Proof. Let $\alpha \in \mathcal{W}^0_{\mathcal{G}_1}$. Since $\mathcal{G}_1$ is solitary for Player 1, we have $(\rho, \alpha) \models \varphi$ for every play ρ of G_1. Now, let σ be a strategy for Player 0 in G. Every play consistent with σ in G is a play in G_1, and therefore also a model of φ with respect to α. Thus, $\alpha \in \mathcal{W}^0_{\mathcal{G}}$.

Let $\alpha \in \mathcal{W}^0_{\mathcal{G}}$ and σ be a winning strategy for Player 0 for $\mathcal{G}$ with respect to α. We expand σ for the arena G_0 by picking an arbitrary successor s' for every $s \in V_1$ (in G) and define $\sigma(ws) = s'$. Every play that is consistent with σ in G_0 is consistent with σ in G. Thus, every play consistent with σ is a model of φ with respect to α. Hence, σ is a winning strategy for Player 0 in $\mathcal{G}_0$ with respect to α. Thus, $\alpha \in \mathcal{W}^0_{\mathcal{G}_0}$.

The second claim follows from the first one and Lemma 5.12 (i). □

In the next section, we show that emptiness, universality, and finiteness of $\mathcal{W}_\mathcal{G}^i$ are decidable, if $\mathcal{G}$ is unipolar and solitary. Combining this with Lemma 5.18, we obtain some necessary and some sufficient conditions for the emptiness and universality of $\mathcal{W}_\mathcal{G}^i$ for unipolar two-player PLTL Games.

Also, we have seen in Corollary 5.14 that the emptiness of $\mathcal{W}_\mathcal{G}^0$ and the universality of $\mathcal{W}_\mathcal{G}^1$ for a PLTL$_\mathbf{G}$ game and the emptiness of $\mathcal{W}_\mathcal{G}^1$ and the universality of $\mathcal{W}_\mathcal{G}^0$ for a PLTL$_\mathbf{F}$ game are decidable. This gives even stronger conditions.

5.2.1 Solitary Unipolar PLTL Games

In the following subsection, we deal with the emptiness, universality, and finiteness of the sets $\mathcal{W}_\mathcal{G}^i$ for solitary unipolar PLTL Games. There are several combinations of the Player i', in whose arena the game is played, the Player i, in whose set $\mathcal{W}_\mathcal{G}^i$ we are interested in, the type of winning condition, either PLTL$_\mathbf{G}$ or PLTL$_\mathbf{F}$, and the problem, one of the three mentioned above. Here, Lemma 5.12 comes in handy, since we can solve a great deal of combinations by the various dualities of the sets $\mathcal{W}_\mathcal{G}^i$. We have already seen in Corollary 5.14 that the closure properties of $\mathcal{W}_\mathcal{G}^i$ (for unipolar games) imply a simple solution to some of the problems. For the other cases, it suffices to consider solitary games for Player 0: we show that the $\mathcal{W}_\mathcal{G}^0$ for PLTL$_\mathbf{G}$ Games are also upwards-closed above a certain bound, in addition to being downwards-closed. Dually, the $\mathcal{W}_\mathcal{G}^0$ for PLTL$_\mathbf{F}$ Games are downwards-closed above a certain bound, in addition to being upwards-closed. Building on these additional closure properties, it can be shown that there is a single valuation that determines the universality of $\mathcal{W}_\mathcal{G}^0$ for a PLTL$_\mathbf{G}$ game respectively the emptiness of $\mathcal{W}_\mathcal{G}^0$ for a PLTL$_\mathbf{F}$ game.

In a solitary game for Player 0, a strategy determines a unique play that is consistent with that strategy. Similarly, every play determines a unique strategy for Player 0. Thus, we can reason about plays instead of strategies. For PLTL$_\mathbf{G}$ Games we show that if Player 0 wins with respect to a large fixed α, then she also wins for all β that are even larger. Thus, $\mathcal{W}_\mathcal{G}^0$ is also upwards-closed above a certain bound that depends only on the game. Let $\mathbf{G}_{\leq y}\psi$ be a subformula of φ. The key idea is that if $\alpha(y)$ is large, then ψ holds for a long period of time after every position where $\mathbf{G}_{\leq y}\psi$ holds. We find a loop in that period that can be repeated, thereby prolonging the time that ψ holds. However, we have to make sure that the truth of other subformulae does not change. Lemma 5.19 proves that we can always find such a loop, if α is just big enough. The reasoning for PLTL$_\mathbf{F}$ Games is dual: we show that $\mathcal{W}_\mathcal{G}^0$ is downwards-closed above a bound. Let $\mathbf{F}_{\leq x}\psi$ be a subformula of φ. Instead of repeating loops, we delete loops, thereby shortening the waiting time till ψ holds, for every position where $\mathbf{F}_{\leq x}\psi$ holds. We show that the valuations corresponding to those bounds determine the universality respectively the emptiness of $\mathcal{W}_\mathcal{G}^0$.

We begin by stating the two technical lemmata, one for PLTL$_\mathbf{G}$ Games, and the other for PLTL$_\mathbf{F}$ Games, each followed by a corollary solving the universality respectively emptiness problem. Then we state the main theorems of this chapter. The first one wraps up the discussion about the emptiness, universality, and finiteness problem for solitary unipolar PLTL Games while the second shows that the solutions to various optimization problems regarding the sets $\mathcal{W}^i_{\mathcal{G}}$ for solitary unipolar games can be computed effectively.

In the following, $n \in \mathbb{N}$ is called a *position* of an infinite play $\rho = \rho_0\rho_1\rho_2 \ldots$, and is used to denote ρ_n as well. An *interval* is a subset $\{n, n+1, \ldots, n+l\}$ of $\mathbb{N}$, and its length is $l+1$. Often, we identify infixes $\rho_n \ldots \rho_{n+l}$ of ρ with the interval induced by its positions.

A technical Lemma for solitary $\text{PLTL}_{\mathbf{G}}$ Games

Let $\mathcal{G}_{\mathbf{G}}$ be a solitary $\text{PLTL}_{\mathbf{F}}$ Game. To deal with the universality problem for $\mathcal{W}^0_{\mathcal{G}_{\mathbf{G}}}$ we show that there exists a valuation α that depends only on $\mathcal{G}_{\mathbf{G}}$ such that $\alpha \in \mathcal{W}^0_{\mathcal{G}_{\mathbf{G}}}$ iff $\mathcal{W}^0_{\mathcal{G}_{\mathbf{G}}}$ is universal. This is also the key to the solution of the finiteness problem. We show $\alpha \in \mathcal{W}^0_{\mathcal{G}_{\mathbf{G}}}$ implies $\beta \in \mathcal{W}^0_{\mathcal{G}_{\mathbf{G}}}$ for all $\beta \geq \alpha$. Thus, if $\alpha \in \mathcal{W}^0_{\mathcal{G}_{\mathbf{G}}}$, then $\mathcal{W}^0_{\mathcal{G}_{\mathbf{G}}}$ is also upwards-closed above α. Combining the two types of closure shows that $\mathcal{W}^0_{\mathcal{G}_{\mathbf{G}}}$ is universal, iff $\alpha \in \mathcal{W}^0_{\mathcal{G}_{\mathbf{G}}}$. So, α determines the universality of $\mathcal{W}^0_{\mathcal{G}_{\mathbf{G}}}$.

It remains to show that such a valuation α exists. Since $\mathcal{G}_{\mathbf{G}}$ is a solitary game for Player 0, a strategy σ determines a single play $\rho(s_0, \sigma)$ that is consistent with σ. If $\alpha(y)$ is large, then every position of $\rho(s_0, \sigma)$, where a subformula $\mathbf{G}_{\leq y}\psi$ holds, is followed by a long interval in which ψ holds at every position. If this interval is not as long as $\beta(y)$, then we find a loop in that interval, which Player 0 can repeat while maintaining the satisfaction of the winning condition with respect to β now. We begin the proof by some simplifications, then construct the new strategy, which is just a single play, and prove the correctness of the construction. This proof is an adaption of a proof from [1].

Lemma 5.19. *Let $\mathcal{G}_{\mathbf{G}} = (G, s_0, \varphi)$ be a solitary* $\text{PLTL}_{\mathbf{G}}$ *Game for Player* 0*, and α and β valuations such that $\beta(y) \geq \alpha(y) \geq 2|G|c_\varphi k_\varphi 2^{n_\varphi}$ for all parameters $y \in \text{var}(\varphi)$. Then, $\alpha \in \mathcal{W}^0_{\mathcal{G}_{\mathbf{G}}}$ implies $\beta \in \mathcal{W}^0_{\mathcal{G}_{\mathbf{G}}}$.*

Proof. If φ does not contain a temporal operator parameterized by a variable, then the claim is trivially true. So, in the following we can assume that there is at least one variable in φ.

Since $\mathcal{G}$ is a solitary game for Player 0, it suffices to prove the following: *if there exists a play ρ of G such that $(\rho, \alpha) \models \varphi$, then there exists a play ρ' of G such that $(\rho', \beta) \models \varphi$.* Without loss of generality we can assume that every variable y occurs at most once in φ. If not, rename the other occurrence to some fresh variable y' and expand α and β by $\alpha(y') = \alpha(y)$ and $\beta(y') = \beta(y)$. Now, given a variable z of φ, define $\alpha_z(z) = \alpha(z) + 1$ and $\alpha_z(y) = \alpha(y)$ for all $y \neq z$. The valuation β can be obtained from α by a sequence of α_z. Thus, we can reformulate our statement again: *if there exists a play ρ of G such that $(\rho, \alpha) \models \varphi$, then there exists a play ρ' of G such that $(\rho', \alpha_z) \models \varphi$.*

Suppose $\mathbf{G}_{\leq z}\psi_z$ is the subformula indexed with z. The crucial case is a position n of ρ where $\mathbf{G}_{\leq z}\psi_z$ holds, but ψ_z holds only at the next $\alpha(z)$ positions, but not $\alpha(z)+1$ positions. Our goal is to repeat a loop of ρ such that ψ_z holds for more than $\alpha(z)$ positions in the resulting play. However, we have to ensure that all other subformulae are satisfied by the new play. Therefore, we only repeat loops such that the same subformulae of φ hold at the first respectively last position of the loop and the subformulae $G_{\leq y}\psi_y$ do not change their truth value throughout the loop. The temporal operators parameterized with constants have to be treated special, to make sure that the bound is not violated by repeating the loop. Therefore, we have to add additional formulae to the set of subformulae: for a PLTL$_\mathbf{G}$ formula $\varphi_\mathbf{G}$ define the *closure of* φ $\mathrm{cl}(\varphi_\mathbf{G})$ inductively by

- $\mathrm{cl}(p) = \{p\}$, and $\mathrm{cl}(\neg p) = \{\neg p\}$,
- $\mathrm{cl}(\gamma \wedge \delta) = \{\gamma \wedge \delta\} \cup \mathrm{cl}(\gamma) \cup \mathrm{cl}(\delta)$,
- $\mathrm{cl}(\gamma \vee \delta) = \{\gamma_1 \vee \delta\} \cup \mathrm{cl}(\gamma) \cup \mathrm{cl}(\delta)$,
- $\mathrm{cl}(\mathbf{X}\gamma) = \{\mathbf{X}\gamma\} \cup \mathrm{cl}(\gamma)$,
- $\mathrm{cl}(\gamma\mathbf{U}\delta) = \{\gamma\mathbf{U}\delta\} \cup \mathrm{cl}(\gamma) \cup \mathrm{cl}(\delta)$,
- $\mathrm{cl}(\gamma\mathbf{R}\delta) = \{\gamma\mathbf{R}\delta\} \cup \mathrm{cl}(\gamma) \cup \mathrm{cl}(\delta)$,
- $\mathrm{cl}(\mathbf{F}_{\leq c}\gamma) = \{\mathbf{F}_{\leq d} \mid d \in [c]\} \cup \mathrm{cl}(\gamma)$,
- $\mathrm{cl}(\mathbf{G}_{\leq c}\gamma) = \{\mathbf{G}_{\leq d} \mid d \in [c]\} \cup \mathrm{cl}(\gamma)$, and
- $\mathrm{cl}(\mathbf{G}_{\leq y}\gamma) = \{\mathbf{G}_{\leq y}\gamma\} \cup \mathrm{cl}(\gamma)$,

A subset $C \subseteq \mathrm{cl}(\varphi_\mathbf{G})$ is called *monotone*, if $\mathbf{F}_{\leq d}\gamma \in C$ implies $\mathbf{F}_{\leq d'}\gamma \in C$ for all $d < d'$ and $\mathbf{G}_{\leq d}\gamma \in C$ implies $\mathbf{G}_{\leq d'}\gamma \in C$ for all $d' < d$. This reflects the monotonicity of the parameterized operators. The number of monotone subsets of $\mathrm{cl}(\varphi)$ can be bounded by $c_\varphi 2^{n_\varphi}$. For a position n of ρ let

$$\pi_n = \{\psi \in \mathrm{cl}(\varphi) \mid (\rho, n, \alpha) \models \psi\}.$$

Every π_n is monotone.

For $y \in \mathrm{var}(\varphi)$ let $\mathbf{G}_{\leq y}\psi_y$ the unique subformula parameterized by y. A position n of ρ is *y-critical*, if $\mathbf{G}_{\leq y}\psi_y \in \pi_n$, but $\mathbf{G}_{\leq y}\psi_y \notin \pi_{n+1}$. Then, π_{n+k} contains ψ_y for all $k \leq \alpha(z)$, but $\psi_y \notin \pi_{n+\alpha(z)+1}$. The interval $\{n+1, \ldots, n+\alpha(z)\}$ is an *y-critical interval.* Some simple facts about y-critical intervals I: we have $\mathbf{G}_{\leq y}\psi_y \notin \pi_n$ for all $n \in I$. Also, if $\mathbf{G}_{\leq y}\psi_y \in \pi_n$, then it holds also for all positions $n+k$ until the beginning of an y-critical interval is reached.

So, if $\alpha(z)$ is large enough, we find two positions in every z-critical interval such that their vertices and the set of satisfied subformulae are equal. The next lemma deals with the operators parameterized with variables.

Lemma 5.20. *Every z-critical interval I contains a subinterval J of length $|G|c_\varphi 2^{n_\varphi}+1$ such that for every $y \in \mathcal{Y}$, either $\mathbf{G}_{\leq y}\psi_y$ holds at all positions in J or $\mathbf{G}_{\leq y}\psi_y$ holds at no position in J.*

Proof. Since $\alpha(z) \geq 2|G|c_\varphi k_\varphi 2^{n_\varphi} = : d$, we know that the length of I is at least d. Let I' denote the initial interval of I of length d. By the facts from above, the truth value of a formula $\mathbf{G}_{\leq y}\psi_y$ can change at most twice in I'. If $\mathbf{G}_{\leq y}\psi_y \in \pi_n$, but $\mathbf{G}_{\leq y}\psi_y \notin \pi_{n+1}$, then $n+1$ is the beginning of an y-critical interval, which is at least as long as I' and does not contain a position where $\mathbf{G}_{\leq y}\psi_y$ holds. Since there are $k_\varphi - 1$ variables $y \neq z$ in φ such that their parameterized subformula could change its truth value, there are at most $2(k_\varphi - 1)$ changes in I', which split I' in at most $2k_\varphi - 1$ subintervals without changes, of which one's length is at least $\frac{d}{2k_\varphi - 1} \geq |G|c_\varphi 2^{n_\varphi} + 1$. □

π_n is a monotone subset of $\mathrm{cl}(\varphi)$, and there are at most $c_\varphi 2^{n_\varphi}$ monotone subsets. Thus, we can find two positions $n_b < n_e$ in every z-critical interval such that

- $\pi_{n_b} = \pi_{n_e}$,
- the vertices ρ_{n_b} and ρ_{n_e} are equal,
- for all variables $y \in \mathrm{var}(\varphi)$ either $\mathbf{G}_{\leq y}\psi_y \in \pi_k$ for all $n_b \leq k \leq n_e$ or $\mathbf{G}_{\leq y}\psi_y \notin \pi_k$ for all $n_b \leq k \leq n_e$.

The last statement is satisfied by the subinterval J as in Lemma 5.20 and the first two statements by a simple counting argument, which applies to J as well.

$\rho_{n_b+1} \cdots \rho_{n_e}$ forms a non-trivial loop in G, the same subformulae of φ hold at the beginning of the loop respectively at the end of the loop, and the subformulae parameterized by variables do not change their truth values in the loop. So, such a loop is repeated in every critical interval to prolong the truth of ψ_y. For a z-critical interval I of ρ let n_b^I and n_e^I be the smallest positions of I that satisfy the conditions above. We denote $\rho_{n_b^I+1} \cdots \rho_{n_e}^I$ by $\Delta(I)$. Now we are able to define the new play ρ': for every z-critical interval I of ρ, we repeat $\Delta(I)$ once. We call the first occurrence of $\Delta(I)$ the *master* and the second one the *copy*.

To conclude the proof, we have to verify that ρ' has the desired properties. It is a play of G, since we repeat only loops of the original play. It remains to prove that the construction does eliminate all z-critical intervals and preserves the truth of the other subformulae.

Repeating the $\Delta(I)$ induces a mapping $f : \mathbb{N} \to \mathbb{N}$ from the positions of ρ' to the positions of ρ, mapping a position of ρ' to its original position in ρ. This is shown in Figure 5.2.

Lemma 5.21. *Let $\psi \in \mathrm{cl}(\varphi)$. If $(\rho, f(n), \alpha) \models \psi$, then $(\rho', n, \alpha_z) \models \psi$*

Proof. We have $f(n+1) = f(n) + 1$ unless n is the last vertex of a segment that is repeated beginning at vertex $n+1$ in ρ'. In this situation, $f(n+1) = f(n) - k$ for some $k > 1$ (the length of the segment being repeated) and $\pi_{f(n)} = \pi_{f(n)-k-1}$. We proceed by induction over the construction of ψ.

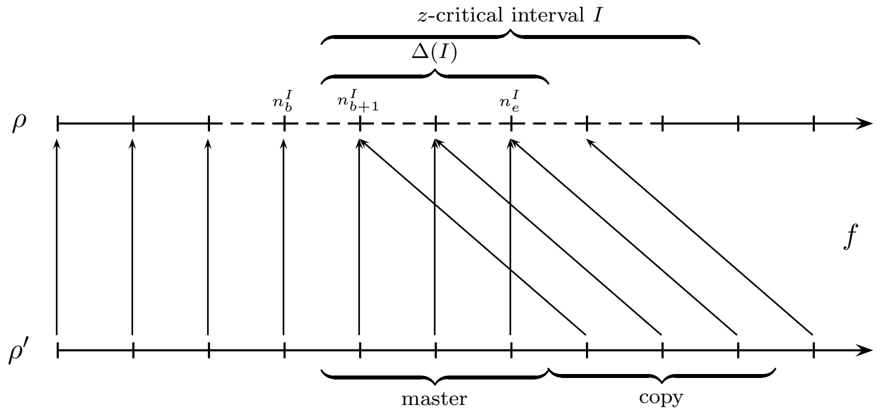

Figure 5.2: The construction of ρ' from ρ. The dashed interval I is z-critical

- The base case, atomic propositions and their negations, is immediate since the vertices at position $f(n)$ in ρ and n in ρ' are equal by construction.
- Conjunctions and disjunctions are inductively true, since they are defined locally.
- $\psi = \mathbf{X}\vartheta$: $(\rho, f(n), \alpha) \models \mathbf{X}\vartheta$, thus $(\rho, f(n)+1, \beta) \models \vartheta$. If $f(n+1) = f(n)+1$, then $(\rho', n+1, \alpha_z) \models \vartheta$ by induction hypothesis, and therefore $(\rho', n, \alpha_z)\mathbf{X} \models \vartheta$.

  If $f(n+1) = f(n) - k$ for some $k > 1$, then $f(n)$ is the last position of a segment $\Delta(I)$ that is repeated in ρ' beginning at position $n+1$. By, construction, we have $\pi_{f(n)-k-1} = \pi_{f(n)}$, thus $(\rho, f(n) - k - 1, \alpha) \models \mathbf{X}\vartheta$ and $(\rho, f(n) - k, \alpha) \models \vartheta$. By $f(n+1) = f(n) - k$ and induction hypothesis $(\rho, n+1, \alpha_z) \models \mathbf{X}\vartheta$.
- $\psi = \gamma\mathbf{U}\vartheta$: $(\rho, f(n), \alpha) \models \gamma\mathbf{U}\vartheta$. Then, there exists a smallest $k \geq 0$ such that $(\rho, f(n)+k, \alpha) \models \vartheta$ and $(\rho, f(n)+l, \alpha) \models \gamma$ for all l such that $0 \leq l < k$.

  If $f(n)$ is not in some segment $\Delta(I)$ that is repeated, then there might be some segments in between the positions $f(n)+1$ and $f(n)+k$ (inclusive) that are repeated, but γ holds at all those positions, so repeating segments does no harm: there is a $k' \geq k$ such that $f(n+k') = f(n)+k$ such that $(\rho, f(n+l), \alpha) \models \gamma$ for all l such that $0 \leq l \leq k'$. Thus, by induction hypothesis $(\rho', n+k', \alpha_z) \models \vartheta$ and $(\rho', n+l, \alpha_z) \models \gamma$ for all l such that $0 \leq l \leq k'$. Hence, $(\rho', n, \alpha_z) \models \gamma\mathbf{U}\vartheta$.

  Now assume that $f(n)$ is in some segment that is repeated, but $f(n)+k$ is in the same segment. Then, $f(n+l) = f(n)+l$ for all l such that $0 \leq l \leq k$ by construction of f. Thus, by induction hypothesis $(\rho', n+k, \alpha_z) \models \vartheta$ and $(\rho', n+l, \alpha_z) \models \gamma$ for all l such that $0 \leq l < k$ and again $(\rho', n, \alpha_z) \models \gamma\mathbf{U}\vartheta$.

  For the last case, assume $f(n)$ is in some $\Delta(I)$ that is repeated and $f(n)+k$ is not in that segment. The case where n is in the copy of $\Delta(I)$ in ρ' is analogous to the first case. Thus, let n be in the master of $\Delta(I)$. Note that there is no position $f(n)+k$ for some $k \geq 0$ in $\Delta(I)$ such that ϑ holds at $f(n)+k$ since k is minimal. Hence, $\gamma\mathbf{U}\vartheta \in \pi_{f(n)+r}$ holds at the last position of $\Delta(I)$. Then, $\gamma\mathbf{U}\vartheta$ holds at the last position before $\Delta(I)$ begins. Thus, it also holds at the first position of Δ, since γ does not hold at the last position before $\Delta(I)$. Now, either there is a prefix of Δ such that this prefix is a (finite) model of $\gamma\mathbf{U}\vartheta$ or γ holds at all positions of $\Delta(I)$. In both cases, we have an $k' \geq 0$ such that ϑ holds at position $f(n+k')$ and γ holds at all positions l' such that $0 \geq l' < l$. Again, by induction hypothesis $(\rho', n, \alpha_z) \models \gamma\mathbf{U}\vartheta$.

- $\psi = \gamma\mathbf{R}\vartheta$: $(\rho, f(n), \alpha) \models \gamma\mathbf{R}\vartheta$, thus for every k either $(\rho, n+k, \alpha) \models \vartheta$ or there exists an $l < k$ and $(\rho, n+l, \alpha) \models \gamma$,

  First, if there exists a (without loss of generality minimal) position $f(n)+k$, for some $k \geq 0$, such that $(\rho, f(n)+k, \alpha) \models \gamma$, then $(\rho, f(n)+l, \alpha) \models \vartheta$ for all l such that $0 \leq l \leq k$. With arguments analogous to the ones for the previous case, we can show that there is a $k' \geq k$ such that $f(n+k') = f(n)+k$ and $(\rho, f(n)+l, \alpha) \models \vartheta$ for all l such that $0 \leq l \leq k'$. Thus, by induction hypothesis $(\rho', n, \alpha_z) \models \gamma\mathbf{R}\vartheta$.

  If there is no position $f(n)+k$, for some $k \geq 0$, such that $(\rho, f(n)+k, \alpha) \models \gamma$, then $(\rho, f(n)+l, \alpha) \models \vartheta$ for all $l \geq 0$. If n is not in the master of some $\Delta(I)$, we immediately conclude $(\rho, f(n+l), \alpha) \models \vartheta$ for all $l \geq 0$, and by induction hypothesis $(\rho', n+l, \alpha_z) \models \vartheta$ for all $l \geq 0$, which implies $(\rho', n, \alpha_z) \models \gamma\mathbf{R}\vartheta$.

  If n is in a master, then we use the fact that $\gamma\mathbf{R}\vartheta$ holds at the last position of $\Delta(I)$ and therefore also at the last position before $\Delta(I)$ begins. Since γ does not hold at that position, $\gamma\mathbf{R}\vartheta$ also holds at the first position of $\Delta(I)$. Thus, there is either a prefix of of Δ such that γ holds at the last position and ϑ at all positions of the prefix or ϑ holds at all positions of $\Delta(I)$.

  In the first case, let $k > 0$ be maximal with $(\rho, f(n)-k, \alpha) \models \gamma$ such that $f(n)-k$ is still in $\Delta(I)$. There is a k' such that $f(n+k') = f(n)-k$ and $(\rho, f(n+l), \alpha) \models \vartheta$ for all l such that $0 \leq l \leq k'$. Thus, by induction hypothesis $(\rho', n, \alpha_z) \models \gamma\mathbf{R}\vartheta$.

  In the second case, we can directly conclude $(\rho, f(n+l), \alpha) \models \vartheta$ for all $l \geq 0$. Then, as above, $(\rho', n, \alpha_z) \models \gamma\mathbf{R}\vartheta$

- $\psi = \mathbf{G}_{\leq c}\vartheta$ for $c \in \mathbb{N}$: let $(\rho, f(n), \alpha) \models \mathbf{G}_{\leq c}\vartheta$. We show $(\rho, f(n+l), \alpha) \models \mathbf{G}_{\leq c-l}\vartheta$ for all $l \leq c$ by induction. Then, $(\rho, f(n+l), \alpha) \models \vartheta$ and $(\rho', n+l, \alpha_z) \models \vartheta$ for all $l \leq c$, by induction hypothesis, which means $(\rho', n, \alpha_z) \models \mathbf{G}_{\leq c}\vartheta$.

  The base case $l = 0$ is clear by the choice of $f(n)$. Now, consider $l > 0$. If $f(n+l) = f(n+l-1)+1$, then $(\rho, f(n+l-1), \alpha) \models \mathbf{G}_{\leq c-(l-1)}\vartheta$ directly implies $(\rho, f(n+l), \alpha) \models \mathbf{G}_{\leq c-l}\vartheta$. If $f(n+l) = f(n+l-1)-k$ for some $k > 0$, i.e., $f(n+l-1)$ is the last position of a segment that is repeated beginning at the next position, we have $\pi_{f(n+l-1)} = \pi_{f(n+l-1)-k-1}$. Thus, $(\rho, f(n+l-1)-k-1, \alpha) \models \mathbf{G}_{\leq c-(l-1)}\vartheta$ and therefore $(\rho, f(n+l-1)-k, \alpha) \models \mathbf{G}_{\leq c-l)}\vartheta$.

- $\psi = \mathbf{F}_{\leq c}\vartheta$ for $c \in \mathbb{N}$: let $(\rho, f(n), \alpha) \models \mathbf{F}_{\leq c}\vartheta$. Analogously to the previous case, we show that for all $l \leq c$ either $(\rho, f(n+l'), \alpha) \models \vartheta$ for some $l' \leq l$ or $(\rho, f(n+l), \alpha) \models \mathbf{F}_{\leq c-l}\vartheta$. Then, there is some $k \leq c$ such that $(\rho, f(n+k), \alpha) \models \vartheta$, thus, by induction hypothesis $(\rho', n+k, \alpha_z) \models \vartheta$ and also $(\rho', n, \alpha_z) \models \mathbf{F}_{\leq c}\vartheta$.

  The base case $l = 0$ is again clear. Thus, let $l > 0$ and $(\rho, f(n+l'), \alpha) \not\models \vartheta$ for all $l' \leq l$. If $f(n+l) = f(n+l-1)+1$, then we can conclude $(\rho, f(n+l), \alpha) \models \mathbf{F}_{\leq c-l}\vartheta$ from $(\rho, f(n+l-1), \alpha) \models \mathbf{F}_{\leq c-(l-1)}\vartheta$ and $(\rho, f(n+l), \alpha) \not\models \vartheta$.

  If $f(n+l) = f(n+l-1)-k$ for some $k > 0$, then again $\pi_{f(n+l-1)} = \pi_{f(n+l-1)-k-1}$. Thus, $(\rho, f(n+l-1)-k-1, \alpha) \models \mathbf{F}_{\leq c-(l-1)}\vartheta$ and $(\rho, f(n+l-1)-k, \alpha) \not\models \vartheta$, therefore we conclude $(\rho, f(n+l-1)-k, \alpha) \models \mathbf{F}_{\leq c-l)}\vartheta$.

- $\psi = \mathbf{F}_{\leq y}\vartheta$ for $y \in \mathcal{Y}$: the choice of the repeated segment guarantees that the truth value of a formula $\mathbf{F}_{\leq y}\psi_y$ stays the same throughout the segment that is repeated and also coincides with the truth value of the last position before the loop. We distinguish two cases.

  Let $y \neq z$ and $(\rho, f(n), \alpha) \models \mathbf{G}_{\leq y}\vartheta$, i.e., $(\rho, f(n)+l, \alpha) \models \vartheta$ for all $l \leq \alpha(y)$. If n is in the master of some segment $\Delta(I)$, we know that ϑ holds at all positions of this segment, since the truth value of $\mathbf{G}_{\leq y}\vartheta$ does not change throughout $\Delta(I)$. Also, repeating a segment $\Delta(I)$ in between $f(n)$ and $f(n)+\alpha(y)$ does no harm, since ϑ holds at all positions of the segment. Thus, for every $l \leq \alpha(y)$ we have $(\rho, f(n+l), \alpha) \models \vartheta$ and $(\rho', n+l, \alpha_z) \models \vartheta$ by induction hypothesis. Therefore, $(\rho', n+l, \alpha_z) \models \mathbf{F}_{\leq y}\vartheta$.

  Now consider $y = z$ and let $(\rho, f(n), \alpha) \models \mathbf{G}_{\leq z}\vartheta$, i.e., $(\rho, f(n)+l, \alpha) \models \vartheta$ for all $l \leq \alpha(z)$. If $f(n)$ is not z-critical, then $(\rho, f(n)+\alpha(z)+1, \alpha) \models \vartheta$. In this case, there are no segments repeated in between $f(n)$ and $f(n)+\alpha(z)$ and $f(n+l) = f(n)+l$ holds for all $l \leq \alpha(z)+1$. Thus, by induction hypothesis we get $(\rho, n+l, \alpha_z) \models \vartheta$ for all $l \leq \alpha_z(z)$ and $(\rho, n, \alpha_z) \models \mathbf{G}_{\leq z}\vartheta$. If $f(n)$ is z-critical, then $f(n)$ is the beginning of a z-critical interval, of which a non-empty subinterval is repeated. Thus, we again obtain $(\rho, f(n+l), \alpha) \models \vartheta$ for all $l \leq \alpha(z)+1$ and by induction hypothesis $(\rho', n+l, \alpha_z) \models \vartheta$ for all $l \leq \alpha_z(z)$; hence, $(\rho, n, \alpha_z) \models \mathbf{G}_{\leq z}\vartheta$. □

To finish the proof of Lemma 5.19, we apply Lemma 5.21 to φ and $n = 0$. Since $f(0) = 0$ and $(\rho, 0, \alpha) \models \varphi$ by assumption, we obtain $(\rho', 0, \alpha_z) \models \varphi$. This suffices as discussed above. □

Corollary 5.22. *Let $\mathcal{G}_{\mathbf{G}} = (G, s, \varphi)$ be a PLTL$_{\mathbf{G}}$ Game, and $\alpha(y) = 2|G|c_\varphi k_\varphi 2^{n_\varphi}$ for all $y \in \mathrm{var}(\varphi)$. Then, $\mathcal{W}^0_{\mathcal{G}_{\mathbf{G}}}$ is universal iff $\alpha \in \mathcal{W}^0_{\mathcal{G}_{\mathbf{G}}}$.*

Proof. One direction is trivial. Thus, let $\alpha \in \mathcal{W}^0_{\mathcal{G}_\mathbf{G}}$ and β be some valuation. If $\beta \leq \alpha$, then $\beta \in \mathcal{W}^0_{\mathcal{G}_\mathbf{G}}$ by downwards-closure of $\mathcal{W}^0_{\mathcal{G}_\mathbf{G}}$, and if $\beta \geq \alpha$, then $\beta \in \mathcal{W}^0_{\mathcal{G}_\mathbf{G}}$ by Lemma 5.19. If β is incomparable to α, then let $k = \max\{\beta(y) \mid y \in \mathrm{var}(\varphi)\}$ and $\beta_k(y) = k$ for all $y \in \mathrm{var}(\varphi)$. We have $\beta_k \geq \alpha$, which implies $\beta_k \in \mathcal{W}^0_{\mathcal{G}_\mathbf{G}}$ by Lemma 5.19. The downwards-closure of $\mathcal{W}^0_{\mathcal{G}_\mathbf{G}}$ and $\beta_k \geq \beta$ finishes the proof. □

A technical Lemma for solitary PLTL$_\mathbf{F}$ Games

Let $\mathcal{G}_\mathbf{F}$ be a solitary PLTL$_\mathbf{F}$ Game. To deal with the emptiness problem for $\mathcal{W}^0_{\mathcal{G}_\mathbf{F}}$ we show that there exists a valuation β that depends only on $\mathcal{G}_\mathbf{F}$ such that $\beta \in \mathcal{W}^0_{\mathcal{G}_\mathbf{F}}$ iff $\mathcal{W}^0_{\mathcal{G}_\mathbf{F}}$ is non-empty. This result also implies the solution of the universality and finiteness problem for $\mathcal{W}^1_{\mathcal{G}_\mathbf{G}}$ by Lemma 5.12. To this end, we show that $\mathcal{W}^0_{\mathcal{G}_\mathbf{F}}$ is also downwards-closed above β. By combining both closure properties we show that β does determine the emptiness of $\mathcal{W}^0_{\mathcal{G}_\mathbf{F}}$. To prove the downwards-closure, we proceed dually to the proof of Lemma 5.19. Since $\mathcal{G}$ is a solitary game, we have to reason about a single play only, which we manipulate. If ψ holds on that play within the next $\beta(x)$ steps of a position, but not within $\alpha(x)$ steps, then we delete a loop in that interval to get the position where ψ holds closer. Again, we have to take care of the other subformulae, especially making sure that we do not delete loops that contain the only position where a subformula holds. If we delete the only position where p holds, then the subformula $\mathbf{F}p$ will no longer hold before the loop. Again, we begin by simplifying the statement to one that speaks about a single play and a single variable that is decreased. Then, we construct the new play and proof the correctness. A statement similar to the following appears in [1], but no proof is given.

Lemma 5.23. *Let $\mathcal{G} = (G, s, \varphi)$ be a solitary* PLTL$_\mathbf{F}$ *Game for Player* 1*, α and β valuations such that $\beta(x) \geq \alpha(x) \geq (2c_\varphi 2^{n_\varphi} + 1) \cdot (|G| c_\varphi 2^{n_\varphi} + 1)$ for all parameters $x \in \mathrm{var}(\varphi)$. Then $\beta \in \mathcal{W}^0_\mathcal{G}$ implies $\alpha \in \mathcal{W}^0_\mathcal{G}$.*

Proof. If φ does not contain a temporal operator parameterized by a variable, then the claim is trivially true. So, in the following we can assume that there is at least one variable in φ.

$\mathcal{G}$ is a solitary game; hence, it suffices to prove the following: *if there exists a play ρ of G such that $(\rho, \alpha) \models \varphi$, then there exists a play ρ' of G such that $(\rho', \beta) \models \varphi$.* We can assume that every variable x occurs at most once in φ. If not, we rename one occurrence and expand the valuations accordingly. Now, given a variable z of φ, define $\beta(z) = \beta(z) - 1$ and $\beta_z(x) = \beta(x)$ for all $x \neq z$. The valuation α can be obtained from β by a sequence of β_z. Thus, we can reformulate our statement again: *if there exists a play ρ of G such that $(\rho, \beta) \models \varphi$, then there exists a play ρ' of G such that $(\rho', \beta_z) \models \varphi$.* Suppose $\mathbf{G}_{\leq z}\psi_z$ is the subformula indexed with z. The crucial case is a position n of ρ where $\mathbf{F}_{\leq z}\psi_z$ holds, but ψ_z only holds at position $n + \alpha(z)$, but not at any earlier position. Our goal is to delete a loop of ρ such that ψ_z holds in less than $\alpha(z)$ positions in the resulting play. Again, we have to ensure that all other subformulae are satisfied by the new play. But it does not suffice to delete loops where the same subformulae of φ hold at the first respectively last position of the loop, since we might delete a position that satisfies a certain subformula ψ. Now assume $\mathbf{F}\psi$ is another subformula. Then, $\mathbf{F}\psi$ holds at the beginning of the loop and we would guarantee that there is a later position in ρ that satisfies ψ, but there is no guarantee that this position is not deleted, again. Thus, we would keep promising the satisfaction of ψ without ever actually delivering. So, we have to adapt our construction to rule out such a situation. Therefore, we delete only loops for which we can guarantee that every formula, that holds in the deleted loop, also holds at a position that is not deleted. For the parameterized operators, we have to ensure that those alternative positions are close-by.

For a PLTL$_\mathbf{F}$ formula $\varphi_\mathbf{F}$ define the *closure* $\mathrm{cl}(\varphi_\mathbf{F})$ inductively by

- $\mathrm{cl}(p) = \{p\}$, and $\mathrm{cl}(\neg p) = \{\neg p\}$,
- $\mathrm{cl}(\gamma \wedge \delta) = \{\gamma \wedge \delta\} \cup \mathrm{cl}(\gamma) \cup \mathrm{cl}(\delta)$,
- $\mathrm{cl}(\gamma \vee \delta) = \{\gamma_1 \vee \delta\} \cup \mathrm{cl}(\gamma) \cup \mathrm{cl}(\delta)$,
- $\mathrm{cl}(\mathbf{X}\gamma) = \{\mathbf{X}\gamma\} \cup \mathrm{cl}(\gamma)$,
- $\mathrm{cl}(\gamma\mathbf{U}\delta) = \{\gamma\mathbf{U}\delta\} \cup \mathrm{cl}(\gamma) \cup \mathrm{cl}(\delta)$,
- $\mathrm{cl}(\gamma\mathbf{R}\delta) = \{\gamma\mathbf{R}\delta\} \cup \mathrm{cl}(\gamma) \cup \mathrm{cl}(\delta)$,
- $\mathrm{cl}(\mathbf{F}_{\leq c}\gamma) = \{\mathbf{F}_{\leq d} \mid d \in [c]\} \cup \mathrm{cl}(\gamma)$,
- $\mathrm{cl}(\mathbf{G}_{\leq c}\gamma) = \{\mathbf{G}_{\leq d} \mid d \in [c]\} \cup \mathrm{cl}(\gamma)$, and
- $\mathrm{cl}(\mathbf{F}_{\leq x}\gamma) = \{\mathbf{F}_{\leq x}\gamma\} \cup \mathrm{cl}(\gamma)$,

A set $C \subseteq \mathrm{cl}(\varphi_\mathbf{G})$ is called *monotone* if $\mathbf{F}_{\leq d}\gamma \in C$ implies $\mathbf{F}_{\leq d'}\gamma \in C$ for all $d < d'$ and $\mathbf{G}_{\leq d}\gamma \in C$ implies $\mathbf{G}_{\leq d'}\gamma \in C$ for all $d' < d$. The number of monotone subsets of $\mathrm{cl}(\varphi)$ can be bounded by $c_\varphi 2^{n_\varphi}$. For a position n of ρ let

$$\pi_n = \{\psi \in \mathrm{cl}(\varphi) \mid (\rho, n, \beta) \models \psi\}.$$

Every π_n is monotone. Furthermore, a finite union of monotone subsets of $\mathrm{cl}(\varphi)$ is monotone.

For $x \in \mathrm{var}(\varphi)$ let $\mathbf{G}_{\leq x}\psi_x$ be the unique subformula parameterized by x. A position n of ρ is *x-critical*, if $\mathbf{F}_{\leq x}\psi_x \in \pi_n$, but $\psi_x \notin \pi_{n+k}$ for all $0 \leq k \leq \beta(z) - 1$. The interval $\{n, \dots, n + \beta(z) - 1\}$ is an *x-critical interval*.

As there are at most $c_\varphi 2^{n_\varphi}$ monotone subsets of $\mathrm{cl}(\varphi)$, every interval of length $(|G|c_\varphi 2^{n_\varphi} + 1)$ of ρ contains two positions n_1 and n_2 such that $\rho_{n_1} = \rho_{n_2}$ and $\pi_{n_1} = \pi_{n_2}$. For every z-critical interval I we decompose its initial interval of length $(2c_\varphi 2^{n_\varphi} + 1) \cdot (|G|c_\varphi 2^{n_\varphi} + 1)$, into $k := (2c_\varphi 2^{n_\varphi} + 1) \cdot (|G|c_\varphi 2^{n_\varphi} + 1)$ intervals $I_1, \dots, I_k$ of length $(|G|c_\varphi 2^{n_\varphi} + 1)$ and define $\Pi_j = \bigcup_{i \in I_j} \pi_i$ for all $j \in [k]$. Every Π_j is monotone, so by the pigeon-hole principle there is at least one monotone $C \subseteq \mathrm{cl}(\varphi)$ such that there exist $j_1 < j_2 < j_3$ with $\Pi_{j_1} = \Pi_{j_2} = \Pi_{j_3}$.

To sum up, we can find two positions $n_b < n_e$ in every z-critical segment I such that

- $\pi_{n_b} = \pi_{n_e}$,
- the vertices ρ_{n_b} and ρ_{n_e} are equal, and
- for every $\psi \in \mathrm{cl}(\varphi)$ such that $(\rho, n, \beta) \models \psi$ for some $n_b \leq n \leq n_e$, there are two positions k_1 and k_2 of I such that
  - $k_1 < n_b$ and $n_e < k_2$,
  - $k_2 - k_1 \leq (2c_\varphi 2^{n_\varphi} + 1) \cdot (|G|c_\varphi 2^{n_\varphi} + 1)$, and
  - $(\rho, k_1, \beta) \models \psi$ and $(\rho, k_2, \beta) \models \psi$.

Thus, $\rho_{n_b+1} \dots \rho_{n_e}$ forms a non-trivial loop in G, the same subformulae of φ hold at the beginning of the loop respectively at the end of the loop and every subformula that holds at some position of the loop also holds at some other position of the interval not too far away.

For a z-critical segment I of ρ let n_b^I and n_e^I be the smallest positions of I that satisfy the conditions above. We denote $\rho_{n_b^I+1} \dots \rho_{n_e^I}$ by $\Delta(I)$. Now we are able to define the new play ρ': for every z-critical interval I of ρ, we delete $\Delta(I)$. To conclude the proof, we have to verify that ρ' has the desired properties. It is a play of G, since we delete only loops of the original play. It remains to prove that the construction did eliminate all z-critical intervals and preserve the truth of the other subformulae. The deletion of the $\Delta(I)$ induces a mapping $f : \mathbb{N} \to \mathbb{N}$ from the positions of ρ' to the positions of ρ, mapping a position of ρ' to its original position in ρ. This is shown in Figure 5.3.

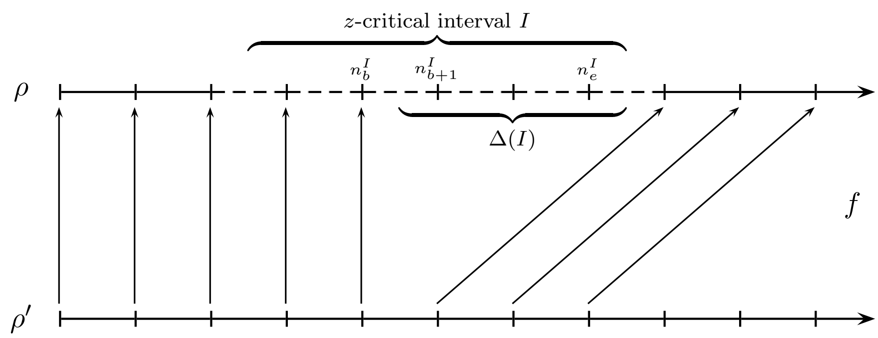

Figure 5.3: The construction of ρ' from ρ. The dashed interval I is z-critical

Lemma 5.24. *Let $\psi \in \mathrm{cl}(\varphi)$. If $(\rho, f(n), \beta) \models \psi$, then $(\rho', n, \beta_z) \models \psi$*

Proof. Notice that $f(n+1) = f(n)+1$ unless $f(n)$ is the last vertex before a deleted loop beginning at position $f(n)+1$. In this situation, $f(n+1) = f(n)+k+1$ for some $k > 0$ (the length of the loop being deleted) and $\pi_{f(n)} = \pi_{f(n)+k}$. We proceed by induction over the construction of ψ.

- The base case, atomic propositions and their negations, is immediate since the vertices at position $f(n)$ in ρ and n in ρ' are equal by construction.
- Conjunctions and disjunctions are inductively true, since they are defined locally.
- $\psi = \mathbf{X}\vartheta$: $(\rho, f(n), \beta) \models \mathbf{X}\vartheta$, thus $(\rho, f(n)+1, \beta) \models \vartheta$. If $f(n+1) = f(n)+1$, then $(\rho', n+1, \beta_z) \models \vartheta$ by induction hypothesis, and therefore $(\rho', n, \beta_z) \models \mathbf{X}\vartheta$.

  If $f(n+1) = f(n)+k+1$ for some $k > 0$, then $f(n)+1$ is the first position of a deleted interval $\Delta(I)$ of length k and $f(n+1) = f(n)+k+1$. By, construction, we have $\pi_{f(n)} = \pi_{f(n)+k}$, thus $(\rho, f(n)+k, \beta) \models \mathbf{X}\vartheta$ and $(\rho, f(n)+k+1, \beta) \models \vartheta$. By $f(n+1) = f(n)+k+1$ and induction hypothesis $(\rho, n+1, \beta_z) \models \mathbf{X}\vartheta$.
- $\psi = \gamma\mathbf{U}\vartheta$: $(\rho, f(n), \beta) \models \gamma\mathbf{U}\vartheta$, thus there exists a smallest $k \geq 0$ such that $(\rho, f(n)+k, \beta) \models \vartheta$ and $(\rho, f(n)+l, \beta) \models \gamma$ for all l such that $0 \leq l < k$.

  If $f(n)+k$ is not in some deleted interval, then $f(n)+k = f(n+k')$ for some $k' \leq k$. By induction hypothesis, we conclude $(\rho', n+k', \beta_z) \models \vartheta$ and $(\rho', n+l', \beta_z) \models \gamma$ for all l' such that $0 \leq l < k'$. Thus, $(\rho', n, \beta_z) \models \gamma\mathbf{U}\vartheta$.

  Now assume that the position $f(n)+k$ is in some deleted interval $\Delta(I)$. Since $(\rho, f(n)+k, \beta) \models \vartheta$, the choice of $\Delta(I)$ guarantees the existence of a position $k' > k$ in $I\backslash\Delta(I)$ such that $(\rho, f(n)+k', \beta) \models \vartheta$. Furthermore, $\gamma\mathbf{U}\vartheta$ holds at the beginning of $\Delta(I)$, thus also at the end. Thus, γ also holds at every position from $f(n)$ to $f(n)+k'-1$, with perhaps the exception of some deleted positions. Thus, $(\rho', n, \beta_z) \models \gamma\mathbf{U}\vartheta$ by induction hypothesis.
- $\psi = \gamma\mathbf{R}\vartheta$: $(\rho, f(n), \beta) \models \gamma\mathbf{R}\vartheta$, thus for every k either $(\rho, n+k, \beta) \models \vartheta$ or there exists l such that $l < k$ and $(\rho, n+l, \beta) \models \gamma$,

  First, assume there is a $k \geq 0$ such that $(\rho, f(n)+k, \beta) \models \gamma$. Let k be minimal with this property. Then, $(\rho, f(n)+l, \beta) \models \vartheta$ for all l such that $0 \leq l \leq k$. With arguments analogous to the ones for the previous case, we can show that there is a $k' \geq k$ that is not deleted, such that $f(n)+k') \models \vartheta$ and $(\rho, f(n)+l, \beta) \models \vartheta$ for all l that are not deleted and such that $0 \leq l \leq k'$. Thus, by induction hypothesis $(\rho', n, \beta_z) \models \gamma\mathbf{R}\vartheta$.

  If there is no position $f(n)+k$, for some $k \geq 0$, such that $(\rho, f(n)+k, \beta) \models \gamma$, then $(\rho, f(n)+l, \beta) \models \vartheta$ for all $l \geq 0$. Now, we immediately conclude $(\rho', n+l, \beta_z) \models \vartheta$ for all $l \geq 0$ by induction hypothesis, which implies $(\rho', n, \beta_z) \models \gamma\mathbf{R}\vartheta$.

- $\psi = \mathbf{G}_{\leq c}\vartheta$ for $c \in \mathbb{N}$: let $(\rho, f(n), \beta) \models \mathbf{G}_{\leq c}\vartheta$. We show $(\rho, f(n+l), \beta) \models \mathbf{G}_{\leq c-l}\vartheta$ for all $l \leq c$ by induction. Then, especially $(\rho, f(n+l), \beta) \models \vartheta$ and thus by induction hypothesis $(\rho', n+l, \beta_z) \models \vartheta$ for all $l \leq c$. Then, $(\rho', n, \beta_z) \models \mathbf{G}_{\leq c}\vartheta$.

  The base case $l = 0$ is clear by the choice of $f(n)$. Now, consider $l > 0$. If $f(n+l) = f(n+l-1)+1$, then $(\rho, f(n+l-1), \beta) \models \mathbf{G}_{\leq c-(l-1)}\vartheta$ directly implies $(\rho, f(n+l), \beta) \models \mathbf{G}_{\leq c-l}\vartheta$.

  If $f(n+l) = f(n+l-1)+k+1$ for some $k > 0$, i.e., $f(n+l-1)$ is the last position before a deleted loop, we have $\pi_{f(n+l-1)} = \pi_{f(n+l-1)+k}$. Therefore, $(\rho, f(n+l-1)+k, \beta) \models \mathbf{G}_{\leq c-(l-1)}\vartheta$ and $(\rho, f(n+l-1)+k+1, \beta) \models \mathbf{G}_{\leq c-l)}\vartheta$, i.e., $(\rho, f(n+l), \beta) \models \mathbf{G}_{\leq c-l)}\vartheta$.

- $\psi = \mathbf{F}_{\leq c}\vartheta$ for $c \in \mathbb{N}$: let $(\rho, f(n), \beta) \models \mathbf{F}_{\leq c}\vartheta$. Analogously to the previous case, we show that for all $l \leq c$ either $(\rho, f(n+l'), \beta) \models \vartheta$ for some $l' \leq l$ or $(\rho, f(n+l), \beta) \models \mathbf{F}_{\leq c-l}\vartheta$. Then, there is some $k \leq c$ such that $(\rho, f(n+k), \beta) \models \vartheta$, thus, by induction hypothesis $(\rho', n+k, \beta_z) \models \vartheta$ and also $(\rho', n, \beta_z) \models \mathbf{F}_{\leq c}\vartheta$.

  The base case $l = 0$ is again clear. Thus, let $l > 0$ and $(\rho, f(n+l'), \beta) \not\models \vartheta$ for all $l' \leq l$. If $f(n+l) = f(n+l-1)+1$, then we can conclude $(\rho, f(n+l), \beta) \models \mathbf{F}_{\leq c-l}\vartheta$ from $(\rho, f(n+l-1), \beta) \models \mathbf{F}_{\leq c-(l-1)}\vartheta$ and $(\rho, f(n+l), \beta) \not\models \vartheta$.

  If $f(n+l) = f(n+l-1)+k+1$ for some $k > 0$, then again $\pi_{f(n+l-1)} = \pi_{f(n+l-1)+k}$. Thus, $(\rho, f(n+l-1)+k, \beta) \models \mathbf{F}_{\leq c-(l-1)}$ and $(\rho, f(n+l-1)+k, \beta) \not\models \vartheta$, therefore we conclude $(\rho, f(n+l-1)+k+1, \beta) \models \mathbf{F}_{\leq c-l)}$, i.e., $(\rho, f(n+l), \beta) \models \mathbf{F}_{\leq c-l)}$.

- $\psi = \mathbf{F}_{\leq x}\psi_x$ for $x \in \mathrm{var}(\varphi)$: let $(\rho, f(n)+k, \beta) \models \mathbf{F}_{\leq x}\psi_x$, thus $(\rho, f(n)+k, \beta) \models \psi_x$ for some $k \leq \beta(x)$.

  First, let $x \neq z$. Notice that $f(n)$ cannot be in a deleted interval by definition of f and that deleting positions in between $f(n)$ and $f(n)+k$ does no harm. In this case, there exists a $k' \leq k$ such that $f(n+k') = f(n)+k$ and $(\rho', n+k', \beta_z) \models \psi_x$ by induction hypothesis, and therefore also $(\rho', n, \beta_z) \models \mathbf{F}_{\leq x}\psi_x$.

  Now assume that the position $f(n)+k$ is in some deleted interval $\Delta(I)$. The choice of $\Delta(I)$ guarantees the existence of two positions k_1 and k_2 of $I \setminus \Delta(I)$ such that

  $$k_1 < k < k_2,\ k_2 - k_1 \leq (2c_\varphi 2^{n_\varphi}+1) \cdot (|G| c_\varphi 2^{n_\varphi}+1) \leq \beta(x),$$

  and ψ_x holds at k_1 and k_2. There is a $l \in \{k_1, k_2\}$ such that $f(n) \leq l \leq f(n)+\beta(x)$: if $k_1 < f(n)$, then

  $$k_2 = k_1 + (k_2 - k_1) < f(n) + (k_2 - k_1) \leq f(n) + \beta(x).$$

Conversely, if $k_2 > f(n) + \beta(x)$, then $k_2 - \beta(x) > f(n)$ and

$$k_1 = k_2 - (k_2 - k_1) \geq k_2 - \beta(x) > f(n).$$

Since the positions are not deleted, there is a $k' \leq \beta(x)$ such that $f(n+k') = f(n)+l$ and we conclude $(\rho', n, \beta_z) \models \mathbf{F}_{\leq x}\psi_x$.

Now consider $x = z$ and let $(\rho, f(n), \beta) \models \mathbf{F}_{\leq z}\vartheta$, i.e., $(\rho, f(n) + k, \beta) \models \vartheta$ for some $k \leq \beta(z)$. If $f(n)$ is not z-critical, then $k < \beta(z)$, i.e., $k \leq \beta(z) - 1 = \beta_z(z)$. Since there are no intervals deleted in between $f(n)$ and $f(n) + \beta(z)$, we have $f(n + l) = f(n) + l$ for all $l \leq \beta(z)$. Thus, by induction hypothesis we get $(\rho, n + k, \beta_z) \models \vartheta$ and therefore $(\rho, n, \beta_z) \models \mathbf{F}_{\leq z}\vartheta$.

If $f(n)$ is z-critical, i.e., $k = \beta(z)$, then $f(n)$ is the beginning of a z-critical interval, of which a non-empty subinterval is deleted. Thus, $f(n+k-1) \leq f(n)+k$ and we obtain $(\rho, f(n+l), \beta) \models \psi_z$ for some $l \leq k-1 = \beta_z(z)$ and by induction hypothesis $(\rho', n + l, \beta_z) \models \psi_z$; hence, $(\rho, n, \beta_z) \models \mathbf{F}_{\leq z}\psi_z$. □

To finish the proof of Lemma 5.23, we apply Lemma 5.24 to φ and $n = 0$. Since $f(0) = 0$ and $(\rho, 0, \beta) \models \varphi$ by assumption, we obtain $(\rho', 0, \beta) \models \varphi$. This suffices as discussed above. □

Corollary 5.25. *Let $\mathcal{G}_\mathbf{F} = (G, s, \varphi)$ be a* PLTL$_\mathbf{F}$ *Game, and $\alpha(x) = (2c_\varphi 2^{n_\varphi} + 1) \cdot (|G|c_\varphi 2^{n_\varphi} + 1)$ for all $y \in \mathrm{var}(\varphi)$. Then, $\mathcal{W}^0_{\mathcal{G}_\mathbf{F}}$ is non-empty iff $\alpha \in \mathcal{W}^0_{\mathcal{G}_\mathbf{F}}$.*

Proof. One direction is trivial. Thus, let $\beta \in \mathcal{W}^0_{\mathcal{G}_\mathbf{F}}$. If $\alpha \geq \beta$, then $\alpha \in \mathcal{W}^0_{\mathcal{G}_\mathbf{F}}$ by upwards-closure of $\mathcal{W}^0_{\mathcal{G}_\mathbf{F}}$, and if $\alpha \leq \beta$, then $\alpha \in \mathcal{W}^0_{\mathcal{G}_\mathbf{F}}$ by Lemma 5.23. If β is incomparable to α, then let $k = \max\{\beta(x) \mid x \in \mathrm{var}(\varphi)\}$ and $\beta_k(x) = k$ for all $x \in \mathrm{var}(\varphi)$. We have $\beta_k \geq \beta$, which implies $\beta_k \in \mathcal{W}^0_{\mathcal{G}_\mathbf{G}}$ by upwards-closure. Since $\alpha \leq \beta$ holds, Lemma 5.23 is applicable and we obtain $\alpha \in \mathcal{W}^0_{\mathcal{G}_\mathbf{F}}$. □

After doing the technical work, we can now reap the fruits of our labor and state the main theorems of this chapter.

Theorem 5.26. *Let $\mathcal{G}$ be a solitary unipolar* PLTL *Game. The emptiness, universality, and finiteness problem for $\mathcal{W}^i_\mathcal{G}$ are decidable.*

Proof. Let $\mathcal{G} = (G, s, \varphi)$ be a solitary PLTL game for Player i'. There are several cases to consider, determined by the identity of the players i and i' (who need not be equal), the type of the winning condition and the problem under consideration. We reduce as many cases as possible to cases already solved, applying the dualities stated in Lemma 5.12. Remember that α_0 is the valuation that maps every variable to zero. Also, the membership problem for $\mathcal{W}^i_\mathcal{G}$ is decidable by solving a single LTL Game with winning condition $\alpha(\varphi)$.

(i) Let $\mathcal{G}$ be solitary for Player 0 and $\varphi \in \text{PLTL}_{\mathbf{G}}$.

- By Corollary 5.14 (i), $\mathcal{W}^0_{\mathcal{G}}$ is non-empty iff it contains α_0.
- For the universality problem of $\mathcal{W}^0_{\mathcal{G}}$ see Corollary 5.22.

(ii) Let $\mathcal{G}$ be solitary for Player 0 and $\varphi \in \text{PLTL}_{\mathbf{F}}$.

- For the emptiness problem of $\mathcal{W}^0_{\mathcal{G}}$ see Corollary 5.25.
- By Corollary 5.14 (ii), $\mathcal{W}^0_{\mathcal{G}}$ is universal iff it contains α_0.

(iii) Let $\mathcal{G}$ be solitary for Player 0 and φ an arbitrary unipolar formula. The emptiness and universality problem for $\mathcal{W}^1_{\mathcal{G}}$ can be decided by considering the complements, again: by Lemma 5.12 (i), $\mathcal{W}^1_{\mathcal{G}}$ is empty iff $\mathcal{W}^0_{\mathcal{G}}$ is universal, and $\mathcal{W}^1_{\mathcal{G}}$ is universal iff $\mathcal{W}^0_{\mathcal{G}}$ is empty. These problems were shown to be decidable in (i) respectively (ii).

(iv) Let $\mathcal{G}$ be solitary for Player 1 and φ an arbitrary unipolar formula. The emptiness and universality problem for $\mathcal{W}^1_{\mathcal{G}}$ can be solved exploiting dualities. By Lemma 5.12 (ii), $\mathcal{W}^1_{\mathcal{G}} = \mathcal{W}^0_{\overline{\mathcal{G}}}$, where $\overline{\mathcal{G}}$ is a solitary game for Player 0 with unipolar winning condition. This is the setting of (i) respectively (ii)

(v) Let $\mathcal{G}$ be solitary for Player 1 and φ an arbitrary unipolar formula. The emptiness and universality problem for $\mathcal{W}^0_{\mathcal{G}}$ can be decided by considering the complements: by Lemma 5.12 (i), $\mathcal{W}^0_{\mathcal{G}}$ is empty iff $\mathcal{W}^1_{\mathcal{G}}$ is universal, and $\mathcal{W}^0_{\mathcal{G}}$ is universal iff $\mathcal{W}^1_{\mathcal{G}}$ is empty. Each of the latter problems is decidable by (iv).

(vi) Now, just the finiteness problems remain. The solution is symmetric: let $\mathcal{G}$ be a solitary game for Player 0 or 1.

- If $\mathcal{G}$ is a $\text{PLTL}_{\mathbf{G}}$ Game, then let $\mathcal{G}_y$ for all $y \in \text{var}(\varphi)$ be defined as for Lemma 5.16. The Lemma states that $\mathcal{W}^0_{\mathcal{G}}$ is infinite iff $W^0_{\mathcal{G}_y}$ is universal for some y. Since $\mathcal{G}_y$ is still a solitary $\text{PLTL}_{\mathbf{G}}$ game we can apply (i) respectively (v).
- If $\mathcal{G}$ is a $\text{PLTL}_{\mathbf{F}}$ Game, then $\mathcal{W}^0_{\mathcal{G}}$ is infinite iff it is non-empty, which was shown to be decidable in (ii) respectively (v).
- For $\mathcal{W}^1_{\mathcal{G}}$ (and both types of winning conditions) we again consider the dual game and obtain $\mathcal{W}^1_{\mathcal{G}} = \mathcal{W}^0_{\overline{\mathcal{G}}}$. The finiteness problem for the latter set was just discussed above. □

As we have reasoned above, the strategy problem for PLTL Games is more of an optimization problem than it is a decision problem. Foremost, a PLTL game is not even a game in the strict sense, since a winning condition φ does not specify the winning plays. Furthermore, for a fixed valuation, the game is equal to an LTL game, a well-known class of games that have been studied extensively. The natural question is to ask for an optimal strategy: given the winning condition $\mathbf{F}_{\leq x} p$, what is the minimal value for x such that Player 0 still has a winning strategy for the instantiated game. Analogously, for the winning condition $\mathbf{G}_{\leq y} p$, what is the maximal value for y such that Player 0 can guarantee a win. We restrict our attention to unipolar games, since in a general winning condition, there are two opposing goals: minimizing the values of the upwards-monotone operators and maximizing the values of the downwards-monotone operators and it is unclear how to resolve this trade-off reasonably.

In the following we show how to find such optimal strategies for solitary unipolar games. The key to the following theorem are again the closure properties of the sets $\mathcal{W}^0_{\mathcal{G}}$. Remember that we restrict the domain of α to the variables occurring in the winning condition.

Theorem 5.27. *Let $\mathcal{G}$ be a solitary unipolar* PLTL *Game. Then, the following optimization problems can be solved effectively.*

(i) If $\mathcal{G}$ is a $\mathrm{PLTL}_{\mathbf{G}}$ *Game: determine* $\max_{\alpha \in \mathcal{W}^0_{\mathcal{G}}} \max_{y \in \mathrm{var}(\varphi)} \alpha(y)$.

(ii) If $\mathcal{G}$ is a $\mathrm{PLTL}_{\mathbf{G}}$ *Game: determine* $\max_{\alpha \in \mathcal{W}^0_{\mathcal{G}}} \min_{y \in \mathrm{var}(\varphi)} \alpha(y)$.

(iii) If $\mathcal{G}$ is a $\mathrm{PLTL}_{\mathbf{F}}$ *Game: determine* $\min_{\alpha \in \mathcal{W}^0_{\mathcal{G}}} \min_{x \in \mathrm{var}(\varphi)} \alpha(x)$.

(iv) If $\mathcal{G}$ is a $\mathrm{PLTL}_{\mathbf{F}}$ *Game: determine* $\min_{\alpha \in \mathcal{W}^0_{\mathcal{G}}} \max_{x \in \mathrm{var}(\varphi)} \alpha(x)$.

Proof. Let $\mathcal{G} = (G, s, \varphi)$ be the game under consideration. If $\mathrm{var}(\varphi) = \emptyset$, then the problems are trivial, so we might as well assume $\mathrm{var}(\varphi) \neq \emptyset$. Also, we can assume that $\mathcal{W}^0_{\mathcal{G}}$ is non-empty, which can be decided by Theorem 5.26. We begin by some general simplifications that are applicable to several cases. These constructions work for arbitrary arenas, not just solitary ones.

If $\varphi \in \mathrm{PLTL}_{\mathbf{G}}$ and $y \in \mathrm{var}(\varphi)$, let φ_y be the formula obtained by replacing every variable $z \neq y$ by zero, and let $\mathcal{G}_y = (G, s, \varphi_y)$. The restriction of $\alpha \in \mathcal{W}^0_{\mathcal{G}}$ to y is contained in $\mathcal{W}^0_{\mathcal{G}_y}$, and for every $\alpha \in \mathcal{W}^0_{\mathcal{G}_y}$, we can expand the domain of α by $\alpha(z) = 0$ for all $z \neq y$. Then, $\alpha \in \mathcal{W}^0_{\mathcal{G}}$. We obtain

$$\max_{\alpha \in \mathcal{W}^0_{\mathcal{G}}} \max_{y \in \mathrm{var}(\varphi)} \alpha(y) = \max_{y \in \mathrm{var}(\varphi)} \max_{\alpha \in \mathcal{W}^0_{\mathcal{G}_y}} \alpha(y). \tag{5.1}$$

Thus, we have reduced the optimization problem to the same problem for a formula with a single variable.

Again, for $\varphi \in \mathrm{PLTL}_{\mathbf{G}}$, let φ' be the formula obtained from φ by renaming every variable to z, and let $\mathcal{G}' = (G, s, \varphi')$. For every $\alpha \in \mathcal{W}^0_{\mathcal{G}}$ define the valuation α' by $\alpha'(z) = \min_{y \in \mathrm{var}(\varphi)} \alpha(y)$. Due to the downwards-closure of $\mathcal{W}^0_{\mathcal{G}}$, we have $\alpha' \in \mathcal{W}^0_{\mathcal{G}'}$. Conversely, for $\alpha' \in \mathcal{W}^0_{\mathcal{G}'}$, define α by $\alpha(y) = \alpha'(z)$ for every $y \in \mathrm{var}(\varphi)$. Then, $\alpha \in \mathcal{W}^0_{\mathcal{G}}$. Again, we obtain

$$\max_{\alpha \in \mathcal{W}^0_{\mathcal{G}}} \min_{y \in \mathrm{var}(\varphi)} \alpha(y) = \max_{\alpha \in \mathcal{W}^0_{\mathcal{G}'}} \alpha(z). \tag{5.2}$$

So, this problem can be reduced to the same problem for a formula with a single variable.

A dual construction is possible for $\varphi \in \mathrm{PLTL}_{\mathbf{F}}$: let φ' be the formula obtained from φ by renaming every variable to z, and let $\mathcal{G}' = (G, s, \varphi')$. Again, for $\alpha \in \mathcal{W}_{\mathcal{G}}$, we define α' by $\alpha'(z) = \max_{x \in \mathrm{var}(\varphi)}$. Due to upwards-closure, we have $\alpha' \in \mathcal{W}^0_{\mathcal{G}'}$. Conversely, for $\alpha' \in \mathcal{W}^0_{\mathcal{G}'}$, we define α by $\alpha(x) = \alpha'(z)$ for all $x \in \mathrm{var}(\varphi)$. Then, $\alpha \in \mathcal{W}^0_{\mathcal{G}}$. Then, we obtain

$$\min_{\alpha \in \mathcal{W}^0_{\mathcal{G}}} \max_{x \in \mathrm{var}(\varphi)} \alpha(x) = \min_{\alpha \in \mathcal{W}^0_{\mathcal{G}'}} \alpha(z), \tag{5.3}$$

which gives a reduction to the problem for a formula with a single variable.

Note that a general construction similar to the one employing $\mathcal{G}_y$ for a $\mathrm{PLTL}_{\mathbf{F}}$ Game is not possible, since there is no equivalent to fixing all but one variable to zero. Minimizing the minimum value turns out to be the most involved case.

The actual proof consists of two steps. First we solve the optimization problems for solitary games for Player 0. Solitary games for Player 1 can then be treated by an easy reduction using the dualities of the sets $\mathcal{W}^i_{\mathcal{G}}$.

- Let $\mathcal{G}$ be a solitary game for Player 0.

  (i) and (ii) By (5.1) and (5.2), it suffices to determine $\max^0_{\mathcal{W}_{\mathcal{G}}} \alpha(y)$, where $\mathcal{G}$ is a $\mathrm{PLTL}_{\mathbf{G}}$ Game whose winning condition φ has a single variable y. Modifying Lemma 5.19 for a formula with a $|\mathrm{var}(\varphi)| = 1$, we can restrict the maximum to

  $$\max_{\alpha \in \mathcal{W}^0_{\mathcal{G}}} \alpha(y) < |G| c_\varphi 2^{n_\varphi} + 1 \quad \text{or} \quad \max_{\alpha \in \mathcal{W}^0_{\mathcal{G}}} \alpha(y) = \infty.$$

  The second case can be tested by Corollary 5.22: We have $\max_{\alpha \in \mathcal{W}^0_{\mathcal{G}_y}} \alpha(y) = \infty$ iff $\mathcal{W}^0_{\mathcal{G}}$ is infinite, which is equivalent to $\mathcal{W}^0_{\mathcal{G}}$ being universal by downwards-closure. If the maximum is not ∞, then we can do binary search in the interval $[0, |G| c_\varphi 2^{n_\varphi}]$, which is correct by downwards-closure. Doing this, we need to determine the winner of at most $\log_2(2|G| c_\varphi 2^{n_\varphi})$ LTL Games.

  (iii) Let $\mathcal{G} = (G, s, \varphi)$ be a $\mathrm{PLTL}_{\mathbf{F}}$ Game, solitary for Player 0, such that $\mathcal{W}^0_{\mathcal{G}}$ is non-empty, and let $n_{\mathcal{G}} = (2c_\varphi 2^{n_\varphi} + 1) \cdot (|G| c_\varphi 2^{n_\varphi} + 1)$. By Corollary 5.25, $\mathcal{W}^0_{\mathcal{G}}$ contains α with $\alpha(x) = n_{\mathcal{G}}$ for all $x \in \mathrm{var}(\varphi)$. Thus,

  $$\min_{\alpha \in \mathcal{W}^0_{\mathcal{G}}} \min_x \alpha(x) \leq n_{\mathcal{G}}.$$

  To determine, whether the minimum is even smaller, the emptiness problem for all $\mathcal{W}^0_{\mathcal{G}_{x,n}}$ has to be solved for all $x \in \mathrm{var}(\varphi)$ and all $n < n_{\mathcal{G}}$, where $\mathcal{G}_{x,n}$ is obtained from $\mathcal{G}$ by replacing the variable x of the winning condition φ by n. The smallest n such that $\mathcal{W}^0_{\mathcal{G}_{x,n}}$ is non-empty for some x, is equal to $\min_{\alpha \in \mathcal{W}^0_{\mathcal{G}_{\mathbf{F}}}} \min_x \alpha(x)$. This

n can be determined in two ways. Either by incrementing n beginning at zero looping through all variables $x \in \text{var}(\varphi)$. Alternatively, a binary search in the interval $[0, n_{\mathcal{G}} - 1]$ can be done for every variable x. In the second approach, the upper bound can be adjusted, if a smaller upper bound was found; however, all variables have to be considered, whereas in the first approach, the search can be terminated, as soon as a non-empty set $\mathcal{W}^0_{\mathcal{G}_{x,n}}$ is discovered.

(iv) Again, by (5.3), it suffices to consider a $\text{PLTL}_{\mathbf{F}}$ Game $\mathcal{G}$ whose winning condition has a single variable x. Lemma 5.23 bounds $\min_{\alpha \in \mathcal{W}^0_{\mathcal{G}}} \alpha(x)$ from above by $n_{\mathcal{G}} = (2c_{\varphi_x} 2^{n_{\varphi_x}} + 1) \cdot (|G| c_{\varphi_x} 2^{n_{\varphi_x}} + 1)$; hence, we can apply binary search, again. This time, the winner of at most $\log_2(n_{\mathcal{G}})$ LTL Games have to be determined.

- Let $\mathcal{G}$ be a solitary game for Player 1.

  (i) and (ii) By (5.1) and (5.2), it suffices to consider a game $\mathcal{G}$ whose winning condition φ has a single variable y: we have

$$\max_{\alpha \in \mathcal{W}^0_{\mathcal{G}}} \alpha(y) = \min_{\alpha \in \mathcal{W}^1_{\mathcal{G}}} \alpha(y) - 1 = \min_{\alpha \in \mathcal{W}^0_{\overline{\mathcal{G}}}} \alpha(y) - 1 = \min_{\alpha \in \mathcal{W}^0_{\overline{\mathcal{G}}}} \min_{y} \alpha(y) - 1$$

  by the closure properties of $\mathcal{W}^0_{\mathcal{G}}$ and $\mathcal{W}^1_{\mathcal{G}}$ and Lemma 5.12. Since $\overline{\mathcal{G}}$ is solitary for Player 0, we know how to minimize the minimum of the values, which also solves our original problem.

  (iii) This case is exceptional as it does not employ a reduction to a solitary game for Player 0. Instead, we proceed as for a solitary game for Player 0: let α be the valuation mapping every variable of φ to $n_{\mathcal{G}} = 2|G| c_{\varphi} k_{\varphi} 2^{n_{\varphi}} + 1$. The complement of $\mathcal{W}^0_{\mathcal{G}}$, the set $\mathcal{W}^1_{\mathcal{G}}$, is not universal by assumption. Thus, it cannot contain α by Lemma 5.19 and Lemma 5.12 (ii). Hence, $\alpha \in \mathcal{W}^0_{\mathcal{G}}$. This gives an upper bound on the value we are interested in. So, we can again check all $\mathcal{W}^0_{\mathcal{G}_{x,n}}$ for non-emptiness, where $\mathcal{G}_{x,n}$ is defined as in case (iii) above.

  (iv) This time, due to (5.3), it suffices to consider a game $\mathcal{G}$ whose winning condition has a single variable x. We have

$$\min_{\alpha \in \mathcal{W}^0_{\mathcal{G}}} \alpha(z) = \max_{\alpha \in \mathcal{W}^1_{\mathcal{G}}} \alpha(x) + 1 = \max_{\alpha \in \mathcal{W}^0_{\overline{\mathcal{G}}}} \alpha(x) + 1 = \max_{\alpha \in \mathcal{W}^0_{\overline{\mathcal{G}}}} \max_{y} \alpha(x) + 1,$$

  by the closure properties of $\mathcal{W}^0_{\mathcal{G}}$ and $\mathcal{W}^1_{\mathcal{G}}$ and Lemma 5.12. The game $\overline{\mathcal{G}}$ is again solitary for Player 0. □

Note that the missing problems, the ones with leading min for $\text{PLTL}_{\mathbf{G}}$ conditions, and the ones with leading max for $\text{PLTL}_{\mathbf{F}}$ conditions are trivial, since these optima are either undefined, if $\mathcal{W}^0_{\mathcal{G}}$ is empty, or they are 0 by downwards-closure respectively ∞ by upwards-closure. Also, the missing optimization problems for $\mathcal{W}^1_{\mathcal{G}}$ can be solved by dualizing the games according to Lemma 5.12 (ii).

Theorem 5.27 only holds for solitary games. However, combining it with Lemma 5.18, one can obtain upper and lower bounds for the optimal strategies in a two-player game. If Lemmata in spirit of Lemma 5.19 and 5.23 hold for unipolar two-player games, then the optimization problems for these games are solvable as well.

Table 5.1 lists the complexity (in the number of LTL Games to solve) of the problems discussed in this section. Solving an LTL Game is **2EXPTIME**-complete for two-player games and **PSPACE**-complete for solitary games in the size of the LTL winning condition. This size is linear in n_φ and the sum of the $\alpha(z)$ and the sum of the constants in φ. Many decision problems can be solved by determining the winner with respect to α_0, which eliminates the influence of α. In all other cases, the values $\alpha(z)$ are a product of $|G|$, c_φ and 2^{n_φ}. Hence, the resulting LTL winning condition might be exponential in n_φ.

G	φ	Problem	LTL Games to solve
arbitrary	$\mathrm{PLTL}_{\mathbf{G}}$	$\mathcal{W}_\mathcal{G}^0$ empty?	1
solitary Pl. 0	$\mathrm{PLTL}_{\mathbf{G}}$	$\mathcal{W}_\mathcal{G}^0$ universal?	1
solitary Pl. 1	$\mathrm{PLTL}_{\mathbf{G}}$	$\mathcal{W}_\mathcal{G}^0$ universal?	1
solitary Pl. 0	$\mathrm{PLTL}_{\mathbf{G}}$	$\mathcal{W}_\mathcal{G}^0$ finite?	k_φ
solitary Pl. 1	$\mathrm{PLTL}_{\mathbf{G}}$	$\mathcal{W}_\mathcal{G}^0$ finite?	k_φ
solitary Pl. 0	$\mathrm{PLTL}_{\mathbf{G}}$	$\max_{\alpha\in\mathcal{W}_\mathcal{G}^0} \max_{y\in\mathrm{var}(\varphi)} \alpha(y)$	$k_\varphi \cdot \log_2(2\|G\|c_\varphi 2^{n_\varphi})$
solitary Pl. 1	$\mathrm{PLTL}_{\mathbf{G}}$	$\max_{\alpha\in\mathcal{W}_\mathcal{G}^0} \max_{y\in\mathrm{var}(\varphi)} \alpha(y)$	$2k_\varphi^2 \cdot \log_2(2\|G\|c_\varphi 2^{n_\varphi}+1)$
solitary Pl. 0	$\mathrm{PLTL}_{\mathbf{G}}$	$\max_{\alpha\in\mathcal{W}_\mathcal{G}^0} \min_{y\in\mathrm{var}(\varphi)} \alpha(y)$	$\log_2(2\|G\|c_\varphi 2^{n_\varphi})$
solitary Pl. 1	$\mathrm{PLTL}_{\mathbf{G}}$	$\max_{\alpha\in\mathcal{W}_\mathcal{G}^0} \min_{y\in\mathrm{var}(\varphi)} \alpha(y)$	$2 \cdot \log_2(2\|G\|c_\varphi 2^{n_\varphi}+1)$
arbitrary	$\mathrm{PLTL}_{\mathbf{F}}$	$\mathcal{W}_\mathcal{G}^0$ universal?	1
solitary Pl. 0	$\mathrm{PLTL}_{\mathbf{F}}$	$\mathcal{W}_\mathcal{G}^0$ empty?	1
solitary Pl. 1	$\mathrm{PLTL}_{\mathbf{F}}$	$\mathcal{W}_\mathcal{G}^0$ empty?	1
solitary Pl. 0	$\mathrm{PLTL}_{\mathbf{F}}$	$\mathcal{W}_\mathcal{G}^0$ finite?	1
solitary Pl. 1	$\mathrm{PLTL}_{\mathbf{F}}$	$\mathcal{W}_\mathcal{G}^0$ finite?	1
solitary Pl. 0	$\mathrm{PLTL}_{\mathbf{F}}$	$\min_{\alpha\in\mathcal{W}_\mathcal{G}^0} \min_{x\in\mathrm{var}(\varphi)} \alpha(x)$	$2k_\varphi \cdot \log_2(2\|G\|c_\varphi 2^{n_\varphi}+1)$
solitary Pl. 1	$\mathrm{PLTL}_{\mathbf{F}}$	$\min_{\alpha\in\mathcal{W}_\mathcal{G}^0} \min_{x\in\mathrm{var}(\varphi)} \alpha(x)$	$k_\varphi \cdot (2\|G\|c_\varphi k_\varphi 2^{n_\varphi}+1)$
solitary Pl. 0	$\mathrm{PLTL}_{\mathbf{F}}$	$\min_{\alpha\in\mathcal{W}_\mathcal{G}^0} \max_{x\in\mathrm{var}(\varphi)} \alpha(x)$	$\log_2(2\|G\|c_\varphi 2^{n_\varphi}+1)$
solitary Pl. 1	$\mathrm{PLTL}_{\mathbf{F}}$	$\min_{\alpha\in\mathcal{W}_\mathcal{G}^0} \max_{x\in\mathrm{var}(\varphi)} \alpha(x)$	$\log_2(2\|G\|c_\varphi 2^{n_\varphi})$

Table 5.1: Complexity of decision and optimization problems

The optimization results can be applied to solitary Request-Response Games, which can be transformed into $\mathrm{PLTL}_{\mathbf{F}}$ Games as seen in Example 5.9.

Lemma 5.28. *For a solitary Request-Response Game, optimal bounds on the waiting times can be computed effectively.*

5.3 Non-uniform Semantics for PLTL Games

In this section, we consider alternative semantics for games with PLTL winning conditions. Our notion of winning PLTL Games as introduced above employed a fixed, *uniform* valuation α and required that every play is a model of the winning condition with respect to α. Hence, once the valuation is fixed, a PLTL Game is in fact an LTL Game, with all of its consequences, most importantly finite-state determinacy. So, both players know the bounds in advance and can choose their moves based on that knowledge. Player 0 has to enforce the fixed bounds on all plays consistent with her strategy to win a PLTL game with respect to a fixed α. This is rather restrictive. An alternative is to determine the winner of a play ρ non-uniformly: Player 0 wins ρ if there exists some witness α such that ρ is a model of φ with respect to α. Then, different plays might have different α that are witnesses for the win for Player 0. Also, these semantics are no longer symmetric and tend to favor Player 0, since she has to play such that there is some witness while Player 1 has to play against all valuations. Formally, a *non-uniform* PLTL *Game* (G, s, φ) consists of an arena G, initial vertex s of G and winning condition $\varphi \in$ PLTL. Let $\rho = \rho_0\rho_1\rho_2\ldots$ be a play in G. Player 0 wins ρ iff there exists a valuation α such that $(\rho, 0, \alpha) \models \varphi$. This is not equivalent to replacing every parameterized operator by its unparameterized version, since the non-uniform winning condition still requires a fixed bound for every single play.

We begin by analyzing the new semantics.

Theorem 5.29. (G, s, φ) *is determined.*

Proof. For every $\varphi \in$ PLTL and every α, the set

$$L(\alpha(\varphi)) = \{\rho \mid \rho \text{ is a path of } G \text{ and } (\rho, 0, \alpha) \models \varphi\}$$

is a ω-regular language, which is Borel. The set of winning plays for Player 0 in (G, s, φ) is $\bigcup_\alpha L(\alpha(\varphi))$, which is a countable union of Borel Sets and therefore a Borel Set. Thus, (G, s, φ) is determined by Theorem 2.13. □

The next question is whether the players can do better than that, i.e., whether they always have finite-state winning strategies. Positional winning strategies do not suffice, as they do not suffice to win LTL Games, which are subsumed by non-uniform PLTL Games. It is easy to show that Player 1 can not hope for finite-state winning strategies.

Theorem 5.30. *There is a non-uniform* PLTL *game* $\mathcal{G}$ *such that Player* 1 *has a winning strategy for* $\mathcal{G}$*, but not a finite-state winning strategy.*

Proof. Player 0 has to play against all possible valuations α. This observation is the key to defining $\mathcal{G}$: consider $\mathcal{G} = (G, s_0, \mathbf{FG}p \vee \mathbf{GF}_{\leq y}\neg p)$ where G is given in Figure 5.4. Note that $\mathcal{G}$ is a solitary game; hence, Player 1 wins if there is a path ρ such that $\rho \models \alpha(\neg\varphi)$ for all α. This is equivalent to $\rho \models \mathbf{GF}\neg p \wedge \mathbf{FG}_{>k}p$ for all k. Thus, Player 1 has to move the token to s_1 infinitely often, but has to keep it in s_0 for more than k

consecutive moves for every k: for the play $\rho = s_0 s_1 s_0 s_0 s_1 s_0 s_0 s_0 s_1 s_0 s_0 s_0 s_0 \ldots$ we have $(\rho, 0, \alpha) \not\models \mathbf{FG}p \lor \mathbf{GF}_{\leq y} \neg p$ for all valuations α. Thus, Player 1 wins $\mathcal{G}_1$.

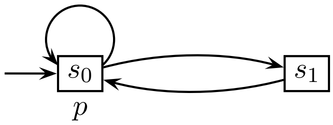

Figure 5.4: The arena G for Theorem 5.30

To complete the proof, assume that Player 1 has a finite-state strategy τ. Since $\mathcal{G}$ is a solitary game, there is exactly one play ρ consistent with τ, which is ultimately periodic by Remark 2.12, i.e., $\rho = xy^\omega$ for some finite play xy. However, every ultimately periodic play is won by Player 0. Thus, Player 1 cannot have a finite-state winning strategy. □

The question for Player 0 remains open. The first idea, to force her to measure the length of some interval and require her to wait even longer at another vertex, but then leaving it, cannot be specified by an PLTL formula, due to the use of distinct variables for upwards-monotone and downwards-monotone operators.

Another important question deals with the relation of classical PLTL games and non-uniform PLTL games. We begin by considering PLTL games and consider their analogons with non-uniform semantics. If Player 0 has a winning strategy for $\mathcal{G}$ with respect to a fixed α, i.e., $\mathcal{W}_\mathcal{G}^0 \neq \emptyset$, then Player 0 wins $\mathcal{G}$ with non-uniform semantics as well. If $\mathcal{W}_\mathcal{G}^0 = \emptyset$, then the situation is different.

Lemma 5.31. *(i) There is a* PLTL *Game such that* $\mathcal{W}_\mathcal{G}^0 = \emptyset$*, but Player* 0 *wins* $\mathcal{G}$ *with non-uniform semantics.*

(ii) There is a PLTL *Game such that* $\mathcal{W}_\mathcal{G}^0 = \emptyset$ *and Player* 1 *wins* $\mathcal{G}$ *with non-uniform semantics.*

Proof. (i) Consider the arena G in Figure 5.5 and the game $(G, s_0, \mathbf{F}_{\leq} p \lor \mathbf{G} \neg p)$. We have $\mathcal{W}_\mathcal{G}^0 = \emptyset$, since Player 0 can keep the token in s_0 for more than $\alpha(y)$ moves and then move it to s_1 for every α, thereby winning the play. On the other hand, for every play ρ in G, there is a valuation α such that $(\rho, 0, \alpha) \models \mathbf{F}_{\leq} p \lor \mathbf{G} \neg p$. Hence, Player 0 wins the game with non-uniform semantics.

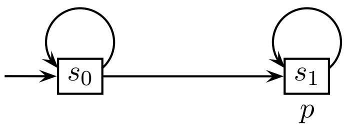

Figure 5.5: The arena G for Lemma 5.31 (i)

(ii) This is trivially true, just pick a game with variable-free winning condition that Player 0 does not win. □

One case of the other direction, from non-uniform semantics to classical semantics, is easy, again. If Player 0 does not win $\mathcal{G}$ with non-uniform semantics, then there is no α such that she wins $\mathcal{G}$ with respect to α. The other case would imply a solution to the question of finite-state determinacy. Assume that the fact that Player 0 wins $\mathcal{G}$ with non-uniform semantics implies $\mathcal{W}^0_{\mathcal{G}} \neq \emptyset$. Then Player 0 can use the finite-state winning strategy σ for $\mathcal{G}$ with respect to some $\alpha \in \mathcal{W}^0_{\mathcal{G}}$ to win $\mathcal{G}$ with non-uniform semantics as well. This gives an alternative way of proving finite-state determinacy (for Player 0) of non-uniform PLTL Games.

Lemma 5.32. *If $\mathcal{G}$ is a win for Player 0 with non-uniform semantics implies $\mathcal{W}^0_{\mathcal{G}} \neq \emptyset$, then Player 0 has a finite-state winning strategy for $\mathcal{G}$ with non-uniform semantics.*

Note that results in spirit of Lemma 5.19 and 5.23 for unipolar two-player games would imply that Player 0 has finite-state winning strategies for unipolar PLTL Games with non-uniform semantics.

Chapter 6

Conclusion

In this work, we investigated the definition of time-optimal strategies based on natural notions of waiting times for several winning conditions for infinite games on graphs. This research is motivated by the fact that these waiting times typically correspond to periods of waiting in the system modeled by the arena. This quality measure of a winning strategy is defined semantically, as opposed to the memory size needed to implement the strategy. Both are important in applications, but historically, attention was only paid to the memory size. Only recently, strategies were evaluated in terms of the quality of the plays it allows. While winning is still a binary notion, i.e., we considered zero-sum games, there are winning plays for Player 0 that are less desirable than others.

The games considered here, Request-Response, Poset, and PLTL Games are characterized by plays that might contain infinitely many (independent) periods of waiting; hence, the waiting times have to be aggregated appropriately. For the former two winning conditions, the limit of the accumulated waiting times is used to aggregate the waiting times. This approach penalizes longer waiting times increasingly stronger, which is desirable in applications. For games with PLTL winning conditions, we employed bounded temporal operators. This amounts to imposing global bounds on the waiting times for eventualities, for example.

For Request-Response Games and Poset Games we proved the existence of time-optimal finite-state winning strategies. The proof technique employed is very flexible and can be applied to other winning conditions as well, if they meet some requirements: waiting times are triggered by a single request and can be computed locally for every step. Also, if the waiting times are high, then the corresponding interval has to contain loops that can be skipped. For solitary unipolar PLTL Games we proved that it is decidable, whether a player wins a game with respect to some, infinitely many, or all valuations. Furthermore, optimal strategies are computable for these games as well. For for two-player unipolar games some of the decision problems are proven to be decidable and necessary and sufficient conditions are given. Lastly, alternative semantics were defined and compared to the standard semantics. Non-uniform PLTL games are determined, but finite-state strategies do no longer suffice, unlike in PLTL Games.

The investigation of time-optimal strategies is far from being finished. We did not only obtain existence results for Request-Response and Poset Games. But the techniques presented to synthesize winning strategies, based on reductions to Mean-Payoff Games, do not allow a sensible implementation, since the memory requirements of the strategies are very high. Also, for PLTL Games, most problems for two-player games are still open. We will conclude this thesis by discussing the open problems just mentioned and by giving some pointers to future research.

Open Questions and Further Research

As mentioned above, the size of the memory structure used to determine time-optimal winning strategies in Request-Response and Poset Games prevents an implementation. However, a careful analysis of the maximal length of non-dickson sequences of waiting time vectors should improve the situation drastically. The following observation is key: if there are $|G|$ moves in which no open request is responded, then this infix contains a loop that is deleted by the strategy improvement operators, as the waiting times increase monotonically. Thus, there is at least one response of a condition in every such infix, which bounds the possible values occurring in these vectors. Furthermore, the waiting time vectors evolve synchronously if they are not reset to zero, which limits the length of a non-dickson sequence. Hence, we believe that the waiting times can be bounded by a smaller bound than $b(n, k)$. If this bound can be lowered sufficiently, then the synthesis algorithm could be implemented. To complement this, one should determine matching lower bounds on the length of a non-dickson sequence of waiting time vectors.

Another interesting aspect is the trade-off between the size of a finite-state strategy and its value. If there is some connection between the two magnitudes, then approximation algorithms might be applicable. These algorithms determine a strategy whose value is only a constant factor higher than the optimal value, but whose memory requirements are lower. Finally, the usefulness of heuristic solutions to the problem of finding time-optimal strategies should be researched.

The most pressing questions for unipolar PLTL Games are the open problems for two-player games, namely, the emptiness, universality, and finiteness problem for $\mathcal{W}_{\mathcal{G}}^0$ and the optimization problems. Counterparts of the technical lemmata proved for solitary games are not directly clear, as there is no longer a single play that has to be manipulated, but the restricted unraveling of the game with respect to a given winning strategy for Player 0. A straightforward idea is to manipulate each play at a time. But when repeating (or deleting) intervals of a single play, one has to pay attention to (possibly infinitely many) other plays that share the prefix that is manipulated. The critical intervals of these plays might overlap in a way that there is no *safe* interval in the common prefix. This problem can be avoided by reasoning about runs of a deterministic automaton accepting exactly the winning plays (with respect to the fixed valuation). However, a run of the automaton cannot have a state repetition in a critical interval, as it has to count the length of the interval. Also, retreating to non-deterministic or alternating automata does not help, since such an automaton might have to be in different states, depending on the continuation of the play. Finally, reasoning with finite-state strategies does not seem helpful, as the size of a strategy is always higher than the length of a critical interval. Hence, there is no (memory) state repetition in a critical interval. The bounds involved in the two technical lemmata do not coincide, which hints at an influence of the size of V_0 (respectively V_1) on these bounds.

Another open problem is the exact complexity of solving the emptiness, universality, and finiteness problem for $\mathcal{W}^0_{\mathcal{G}}$, as well as the complexity of the optimization problems. The results for LTL Games give lower bounds, and the translation of PLTL to LTL (with respect to a fixed valuation), which increases the size of the formula linearly in the values of the variables, gives upper bounds on the complexities.

Lastly, we want to mention an idea related to time-optimal strategies: Muller Games do not have a clear definition of waiting times. Instead, McNaughton investigated the following question [40]: can a referee decide the winner of an infinite play after a finite play of a certain duration, depending on the winning condition and the arena? For positionally determined games, a play can be stopped as soon as a vertex is visited for the second time. McNaughton discusses a scoring function for finite plays and appropriate choices for the duration of the play. However, no formal results are given, as he was interested in devising games to be played by humans. But, as finite-state strategies suffice to win Muller Games, the play can be stopped after a finite time. Then, the referee can analyze the loops of the finite play. However, such a play will have several loops and he has to determine the winner based on the right loops, i.e., a player might not win a play, but is able to force some loops, that could mislead the referee.

Bibliography

[1] Rajeev Alur, Kousha Etessami, Salvatore La Torre, and Doron Peled. Parametric temporal logic for "model measuring". In *ICALP*, pages 159–168, 1999.

[2] Rajeev Alur and Thomas A. Henzinger. Real-time logics: Complexity and expressiveness. *Inf. Comput.*, 104(1):35–77, 1993.

[3] Rajeev Alur and Salvatore La Torre. Deterministic generators and games for LTL fragments. *ACM Trans. Comput. Log.*, 5(1):1–25, 2004.

[4] Henrik Björklund, Sven Sandberg, and Sergei G. Vorobyov. A discrete subexponential algorithm for parity games. In *STACS*, pages 663–674, 2003.

[5] Julius R. Büchi. On a decision method in restricted second order arithmetic. In Ernest Nagel, Patrick Suppes, and Alfred Tarski, editors, *Proceedings of the 1960 International Congress on Logic, Methodology and Philosophy of Science*, pages 1–11. Stanford University Press, June 1962.

[6] Julius R. Büchi and Lawrence H. Landweber. Solving sequential conditions by finite-state strategies. *Transactions of the American Mathematical Society*, 138:295–311, April 1969.

[7] Krishnendu Chatterjee and Thomas A. Henzinger. Finitary winning in ω-regular games. In *TACAS*, pages 257–271, 2006.

[8] Alonzo Church. Applications of recursive arithmetic to the problem of circuit synthesis. In *Summaries of the Summer Institute of Symbolic Logic, Cornell Univ., Ithaca, N.Y.*, volume I, pages 3–50, 1957.

[9] Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Logic of Programs*, pages 52–71, 1981.

[10] Edmund M. Clarke, Orna Grumberg, and Doron Peled. *Model Checking.* MIT Press, 1999.

[11] Luca de Alfaro, Thomas A. Henzinger, and Rupak Majumdar. Discounting the future in systems theory. In *ICALP*, pages 1022–1037, 2003.

[12] Leonard E. Dickson. Finiteness of the odd perfect and primitive abundant numbers with n distinct prime factors. *American Journal of Mathematics*, 35(4):413–422, 1913.

[13] Stefan Dziembowski, Marcin Jurdziński, and Igor Walukiewicz. How much memory is needed to win infinite games? In *LICS*, pages 99–110, 1997.

[14] Andrzej Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae*, 49:129–141, 1961.

[15] Andrzej Ehrenfeucht and Jan Mycielski. Positional strategies for mean payoff games. *International Journal of Game Theory*, 8(2):109–113, June 1979.

[16] E. Allen Emerson and Joseph Y. Halpern. "sometimes" and "not never" revisited: On branching versus linear time. In *POPL*, pages 127–140, 1983.

[17] E. Allen Emerson and Charanjit S. Jutla. Tree automata, μ-calculus and determinacy (extended abstract). In *FOCS*, pages 368–377, 1991.

[18] E. Allen Emerson, Charanjit S. Jutla, and A. Prasad Sistla. On model-checking for fragments of μ-calculus. In *CAV*, pages 385–396, 1993.

[19] E. Allen Emerson, Aloysius K. Mok, A. Prasad Sistla, and Jai Srinivasan. Quantitative temporal reasoning. In *CAV*, pages 136–145, 1990.

[20] Diana Fischer, Erich Grädel, and Łukasz Kaiser. Model checking games for the quantitative μ-calculus. In *STACS*, pages 301–312, 2008.

[21] Roland Fräissé. Sur quelques classifications des systèmes de relations. In *Publications Scientifiques de l'Université d'Alger*, volume 1 of *A*, pages 35–182, 1954.

[22] Hugo Gimbert and Wieslaw Zielonka. When can you play positionally? In *MFCS*, pages 686–697, 2004.

[23] Hugo Gimbert and Wieslaw Zielonka. Games where you can play optimally without any memory. In *CONCUR*, pages 428–442, 2005.

[24] Hugo Gimbert and Wieslaw Zielonka. Deterministic priority mean-payoff games as limits of discounted games. In *ICALP (2)*, pages 312–323, 2006.

[25] Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001]*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.

[26] Yuri Gurevich and Leo Harrington. Trees, automata, and games. In *STOC*, pages 60–65, 1982.

[27] Michael Holtmann and Christof Löding. Memory reduction for strategies in infinite games. In *CIAA*, pages 253–264, 2007.

[28] Florian Horn. Faster algorithms for finitary games. In *TACAS*, pages 472–484, 2007.

[29] Florian Horn, Wolfgang Thomas, and Nico Wallmeier. Optimal strategy synthesis in request-response games. In *ATVA*, pages 361–373, 2008.

[30] Jr James E. Kelley and Morgan R. Walker. Critical-path planning and scheduling. In *IRE-AIEE-ACM '59 (Eastern): Papers presented at the December 1-3, 1959, eastern joint IRE-AIEE-ACM computer conference*, pages 160–173, New York, NY, USA, 1959. ACM.

[31] Marcin Jurdziński. Deciding the winner in parity games is in $\mathbf{UP} \cap \mathbf{co} - \mathbf{UP}$. *Inf. Process. Lett.*, 68(3):119–124, 1998.

[32] Marcin Jurdziński. Small progress measures for solving parity games. In *STACS*, pages 290–301, 2000.

[33] Marcin Jurdziński, Mike Paterson, and Uri Zwick. A deterministic subexponential algorithm for solving parity games. In *SODA*, pages 117–123, 2006.

[34] Alexander S. Kechris. *Classical Descriptive Set Theory (Graduate Texts in Mathematics)*. Springer, January 1995.

[35] Orna Kupferman, Nir Piterman, and Moshe Y. Vardi. From liveness to promptness. In *CAV*, pages 406–419, 2007.

[36] Leslie Lamport. "sometime" is sometimes "not never" - on the temporal logic of programs. In *POPL*, pages 174–185, 1980.

[37] Donald A. Martin. Borel determinacy. *Annals of Mathematics*, 102:363–371, 1975.

[38] Robert McNaughton. Finite-state infinite games. Technical report, Massachusetts Institute of Technology, 1965.

[39] Robert McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and Control*, 9(5):521–530, 1966.

[40] Robert McNaughton. Playing infinite games in finite time. In *A Half-Century of Automata Theory*, pages 73–91, 2000.

[41] Andrzej W. Mostowski. Games with forbidden positions. Technical Report 78, Instytut Matematyki, University of Gdansk, 1991.

[42] John Nash. Non-cooperative games. *The Annals of Mathematics*, 54(2):286–295, September 1951.

[43] John Von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[44] Amir Pnueli. The temporal logic of programs. In *FOCS*, pages 46–57, 1977.

[45] Amir Pnueli and Roni Rosner. On the synthesis of a reactive module. In *POPL*, pages 179–190, 1989.

[46] Arthur Prior. *Past, Present and Future*. Oxford University Press, 1967.

[47] Micheal O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969.

[48] Sven Schewe. Solving parity games in big steps. In *FSTTCS*, pages 449–460, 2007.

[49] A. Prasad Sistla and Edmund M. Clarke. The complexity of propositional linear temporal logics. In *STOC*, pages 159–168, 1982.

[50] Wolfgang Thomas. Automata on infinite objects. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, pages 133–192. MIT Press, 1990.

[51] Wolfgang Thomas. Languages, automata, and logic. In *Handbook of Formal Languages*, pages 389–455. Springer, 1997.

[52] Jens Vöge and Marcin Jurdziński. A discrete strategy improvement algorithm for solving parity games. In *CAV*, pages 202–215, 2000.

[53] Nico Wallmeier. *Strategien in unendlichen Spielen mit Liveness-Gewinnbedingungen: Syntheseverfahren, Optimierung und Implementierung*. PhD thesis, RWTH Aachen University, 2008.

[54] Nico Wallmeier, Patrick Hütten, and Wolfgang Thomas. Symbolic synthesis of finite-state controllers for request-response specifications. In *CIAA*, pages 11–22, 2003.

[55] Uri Zwick and Mike Paterson. The complexity of mean payoff games on graphs. *Theor. Comput. Sci.*, 158(1&2):343–359, 1996.

Index

Symbol Index