Fundamenta Informaticae XXI (201X) 1–26
DOI 10.3233/FI-2012-0000
IOS Press

On Promptness in Parity Games*†

Fabio Mogavero

Aniello Murano

Loredana Sorrentino

Università degli Studi di Napoli Federico II

Abstract. *Parity games* are infinite-duration two-player turn-based games that provide powerful formal-method techniques for the automatic synthesis and verification of distributed and reactive systems. This kind of game emerges as a natural evaluation technique for the solution of the μ-calculus model-checking problem and is closely related to alternating ω-automata. Due to these strict connections, parity games are a well-established environment to describe *liveness properties* such as "every request that occurs infinitely often is eventually responded". Unfortunately, the classical form of such a condition suffers from the strong drawback that there is no bound on the effective time that separates a request from its response, i.e., responses are *not promptly* provided. Recently, to overcome this limitation, several variants of parity game have been proposed, in which quantitative requirements are added to the classic qualitative ones. In this paper, we make a general study of the concept of promptness in parity games that allows to put under a unique theoretical framework several of the cited variants along with new ones. Also, we describe simple polynomial reductions from all these conditions to either Büchi or parity games, which simplify all previous known procedures. In particular, they allow to lower the complexity class of *cost* and *bounded-cost parity games* recently introduced. Indeed, we provide solution algorithms showing that determining the winner of these games is in UPTIME $\cap$ COUPTIME.

Keywords: Parity games, formal verification, liveness, promptness, cost-parity games, quantitative games, UPTIME $\cap$ COUPTIME

Address for correspondence: Dip. Ingegneria elettrica e Tecnologie dell'Informazione, Via Claudio, 21, – 80125 Napoli, Italy.
*This paper is an extended version of the paper [27] appeared in LPAR 2013.
†Partially supported by FP7 European Union project 600958-SHERPA, IndAM 2013 project "Logiche di Gioco Estese", Embedded System Cup Project, B25B09090100007 (POR Campania FSE 2007/2013, asse IV e asse V), Italian Ministry of University and Research, and EU under the PON OR.C.HE.S.T.R.A. project.

1. Introduction

Parity games [16, 31] are abstract infinite-duration two-player turn-based games, which represent a powerful mathematical framework to analyze several problems in computer science and mathematics. Their importance is deeply related to the strict connection with other games of infinite duration, in particular, *mean* payoff, *discounted* payoff, *energy*, *stochastic*, and *multi-agent* games [7, 9, 10, 12, 13]. In the basic setting, parity games are played on directed graphs whose nodes are labeled with priorities (namely, *colors*) and players have perfect information about the adversary moves. The two players, player $\exists$ and player $\forall$, move in turn a token along the edges of the graph starting from a designated initial node. Thus, a play induces an infinite path and player $\exists$ wins the play if the greatest priority that is visited infinitely often is even. In the contrary case, it is player $\forall$ that wins the play. An important aspect of parity games is its *memoryless determinacy*, that is, either player $\exists$ or player $\forall$ has a winning strategy, which does not depend on the history of the play [16]. Therefore, such a strategy can be represented as a subset of the edges of the graph and the problem of constructing a winning strategy turns out to be in UPTIME $\cap$ COUPTIME [19]. The question whether or not a polynomial time solution exists for parity games is a long-standing open question.

In formal system design and verification [14, 15, 23, 29], parity games arise as a natural evaluation machinery for the automatic synthesis and verification of distributed and reactive systems [4, 5, 6, 24]. Specifically, in model checking, one can verify the correctness of a system with respect to a desired behavior, by checking whether a model of the system, *i.e.*, a *Kripke structure*, is correct with respect to a formal specification of its behavior, usually described in terms of a logic formula. In case the specification is given as a μ-calculus formula [20], the model checking question can be polynomially rephrased as a parity game [16]. In the years, this approach has been extended to be usefully applied in several complex system scenarios, as in the case of open-systems interacting with an external environment [24], in which the latter has only partial information about the former [25].

Parity games can express several important system requirements such as *safety* and *liveness* properties. Along an infinite play, safety requirements are used to ensure that nothing "bad" will ever happen, while liveness properties ensure that something "good" eventually happens [2]. Often, safety and liveness properties alone are simple to satisfy, while it becomes a very challenging task when properties of this kind need to be satisfied simultaneously. As an example, assume we want to check the correctness of a printer scheduler that serves two users in which it is required that, whenever a user sends a job to the printer, the job is eventually printed out (liveness property) and that two jobs are never printed simultaneously (safety property). The above liveness property can be written as the LTL [28] formula $\mathtt{G}(req \rightarrow \mathtt{F}\ grant)$, where $\mathtt{G}$ and $\mathtt{F}$ stand for the classic temporal operators "always" and "eventually", respectively. This kind of question is also known in literature as a *request-response condition* [18]. As explained above, in a parity game, this requirement is interpreted over an infinite path generated by the interplay of the two players. From a theoretical viewpoint, on checking whether a request is eventually granted, there is no bound on the "waiting time", namely the time elapsed until the job is printed out. In other words, it is enough to check that the system "can" grant the request, while we do not care when it happens. In a real industry scenario, instead, the request is more concrete, *i.e.*, the job must be printed out in a reasonable time bound.

In the last few years, several works have focused on the above timing aspect in system specification. In [22], it has been addressed by forcing LTL to express "prompt" requirements, by means of a *prompt* operator $\mathtt{F}_p$ added to the logic. In [1] the automata-theoretic counterpart of the $\mathtt{F}_p$ operator has been studied. In particular, *prompt-Büchi* automata are introduced and it has been showed that their intersection with ω-regular languages is equivalent to co-Büchi. Successively, the prompt semantics has been lifted to ω-regular games, under the parity winning condition [11], by introducing *finitary parity* games . There, the concept of "*distance*" between positions in a play has been introduced and referred as the number of edges traversed to reach a node from a given one. Then, winning positions of the game are restricted to those occurring bounded. To give few more details, first consider that, as in classic parity games, arenas have vertexes equipped with natural number priorities and in a play every odd number met is seen as a pending "*request*" that, to be satisfied, requires to meet a bigger even number afterwards along the play, which is therefore seen as a "*response*". Then, player $\exists$ wins the game if almost all requests are responded within a bounded distance. It has been shown in [11] that the problem of determining the winner in a finitary parity game is in PTIME.

Recently, the work [11] has been generalized in [17] to deal with more involved prompt parity conditions. For this reason, arenas are further equipped with two kinds of edges, i-*edges* and ϵ-*edges*, which indicate whether there is or not a time-unit consumption while traversing an edge, respectively. Then, the cost of a path is determined by the number of its i-edges. In some way, the cost of traversing a path can be seen as the consumption of resources. Therefore, in such a game, player $\exists$ aims to achieve its goal with a bounded resource, while player $\forall$ tries to avoid it. In particular, player $\exists$ wins a play if there is a bound b such that all requests, except at most a finite number, have a cost bounded by b and all requests, except at most a finite number, are responded. Since we now have an explicit cost associated to every path, the corresponding condition has been named *cost parity* (CP). Note that in cost parity games a finite number of unanswered requests with unbounded cost is also allowed. By disallowing this, in [17], a strengthening of the cost parity condition has been introduced and named *bounded-cost parity* (BCP) condition. There, it has been shown that the winner of both cost parity and bounded-cost parity can be decided in NPTIME $\cap$ CONPTIME.

In this article we keep working on two-player parity games, under the prompt semantics, over colored (vertexes) arenas with or without weights over edges. In the sequel, we refer to the latter as *colored arenas* and to the former as *weighted arenas*. Our aim is twofold. On one side, we give a clear picture of all different extended parity conditions introduced in the literature working under the prompt assumption. In particular, we analyze their main intrinsic peculiarities and possibly improve the complexity class results related to the game solutions. On the other side, we introduce new parity conditions to work on both colored and weighted arenas and study their relation with the known ones. For a complete list of all the conditions we address in the sequel of this article, see Table 1.

In order to make our reasoning more clear, we first introduce the concept of *non-full*, *semi-full* and *full* acceptance parity conditions. To understand their meaning, first consider again the cost parity condition. By definition, it is a conjunction of two properties and in both of them a finite number of requests (possibly different) can be ignored. For this reason, we call this condition "non-full". Consider now the bounded-cost parity condition. By definition, it is still a conjunction of two properties, but now only in one of them a finite number of requests can be ignored. For this reason, we call this condition "semi-full". Finally, a parity condition is named "full" if none of the requests can be ignored. Note that the full concept has been already addressed in [11] on classic (colored) arenas. We also refer to [11] for further motivations and examples.

As a main contribution in this work, we introduce and study three new parity conditions named *full parity* (FP), *prompt parity* (PP) and *full-prompt parity* (FPP) condition, respectively. The full parity condition is defined over colored arenas and, in accordance to the full semantics, it simply requires that all requests must be responded. Clearly, it has no meaning to talk about a semi-full parity condition, as there is just one property to satisfy. Also, the non-full parity condition corresponds to the classic parity one. See Table 2 for a schematic view of this argument. We prove that the problem of checking whether player $\exists$ wins under the full parity condition is in PTIME. This result is obtained by a quadratic translation to classic Büchi games. The prompt parity condition, which we consider on both colored and weighted arenas, requires that almost all requests are responded within a bounded cost, which we name here *delay*. The full-prompt parity condition is defined accordingly. Observe that the main difference between the cost parity and the prompt parity conditions is that the former is a conjunction of two properties, in each of which a possibly different set of finite requests can be ignored, while in the latter we indicate only one set of finite requests to be used in two different properties. Nevertheless, since the quantifications of the winning conditions range on co-finite sets, we are able to prove that prompt and cost parity conditions are semantically equivalent. We also prove that the complexity of checking whether player $\exists$ wins the game under the prompt parity condition is UPTIME $\cap$ COUPTIME, in the case of weighted arenas. So, the same result holds for cost parity games and this improves the previously known results. The statement is obtained by a quartic translation to classic parity games. Our algorithm always reduces the original problem to a unique parity game, which is the core of how we gain a better result (w.r.t. the complexity class point of view). Obviously, this is different from what is done in [17] as the algorithm there performs several calls to a parity game solver. Observe that, on colored arenas, prompt and full-prompt parity conditions correspond to the finitary and bounded-finitary parity conditions [11], respectively. Hence, both the corresponding games can be decided in PTIME. We prove that for full-prompt parity games the PTIME complexity holds even in the case the arenas are weighted. Finally, by means of a cubic translation to classic parity games, we prove that bounded-cost parity over weighted arenas is in UPTIME $\cap$ COUPTIME, which also improves the previously known result about this condition.

Outline The sequel of the paper is structured as follows. In Section 2, we give some preliminary concepts about games. In Section 3, we introduce all parity conditions we successively analyze in Section 4, with respect to their relationships. In Section 5, we show the reductions from cost parity and bounded-cost parity games to parity games in order to prove that they both are in UPTIME $\cap$ COUPTIME. Finally, in the concluding section, we give a complete picture of all complexity results by means of Table 3.

2. Preliminaries

In this section, we describe the concepts of two-player turn-based arena, payoff-arena, and game. As they are common definitions, an expert reader can also skip this part.

2.1. Arenas

An *arena* is a tuple $\mathcal{A} \triangleq \langle \mathrm{Ps}_\exists, \mathrm{Ps}_\forall, Mv \rangle$, where $\mathrm{Ps}_\exists$ and $\mathrm{Ps}_\forall$ are the disjoint sets of *existential* and *universal positions* and $Mv \subseteq \mathrm{Ps} \times \mathrm{Ps}$ is the left-total *move relation* on $\mathrm{Ps} \triangleq \mathrm{Ps}_\exists \cup \mathrm{Ps}_\forall$. The *order* of $\mathcal{A}$ is the number $|\mathcal{A}| \triangleq |\mathrm{Ps}|$ of its positions. An arena is *finite* iff it has finite order. A *path* (resp., *history*) in $\mathcal{A}$ is an infinite (resp., finite non-empty) sequence of vertexes $\pi \in \mathrm{Pth} \subseteq \mathrm{Ps}^\omega$ (resp., $\rho \in \mathrm{Hst} \subseteq \mathrm{Ps}^+$) compatible with the move relation, *i.e.*, $(\pi_i, \pi_{i+1}) \in Mv$ (resp., $(\rho_i, \rho_{i+1}) \in Mv$), for all $i \in \mathbb{N}$ (resp.,

$i \in [0, |\rho| - 1[$), where Pth (resp., Hst) denotes the set of all paths (resp., histories). Intuitively, histories and paths are legal sequences of reachable positions that can be seen, respectively, as partial and complete descriptions of possible outcomes obtainable by following the rules of the game modeled by the arena. An *existential* (resp., *universal*) *history* in $\mathcal{A}$ is just a history $\rho \in \mathrm{Hst}_{\exists} \subseteq \mathrm{Hst}$ (resp., $\rho \in \mathrm{Hst}_{\forall} \subseteq \mathrm{Hst}$) ending in an existential (resp., universal) position, *i.e.*, $\mathsf{lst}(\rho) \in \mathrm{Ps}_{\exists}$ (resp., $\mathsf{lst}(\rho) \in \mathrm{Ps}_{\forall}$). An *existential* (resp., *universal*) *strategy* on $\mathcal{A}$ is a function $\sigma_{\exists} \in \mathrm{Str}_{\exists} \subseteq \mathrm{Hst}_{\exists} \rightarrow \mathrm{Ps}$ (resp., $\sigma_{\forall} \in \mathrm{Str}_{\forall} \subseteq \mathrm{Hst}_{\forall} \rightarrow \mathrm{Ps}$) mapping each existential (resp., universal) history $\rho \in \mathrm{Hst}_{\exists}$ (resp., $\rho \in \mathrm{Hst}_{\forall}$) to a position compatible with the move relation, *i.e.*, $(\mathsf{lst}(\rho), \sigma_{\exists}(\rho)) \in Mv$ (resp., $(\mathsf{lst}(\rho), \sigma_{\forall}(\rho)) \in Mv$), where $\mathrm{Str}_{\exists}$ (resp., $\mathrm{Str}_{\forall}$) denotes the set of all existential (resp., universal) strategies. Intuitively, a strategy is a high-level plan for a player to achieve his own goal, which contains the choice of moves as a function of the histories of the current outcome. A path $\pi \in \mathrm{Pth}(v)$ starting at a position $v \in \mathrm{Ps}$ is the *play* in $\mathcal{A}$ *w.r.t.* a pair of strategies $(\sigma_{\exists}, \sigma_{\forall}) \in \mathrm{Str}_{\exists} \times \mathrm{Str}_{\forall}$ ($((\sigma_{\exists}, \sigma_{\forall}), v)$-*play*, for short) iff, for all $i \in \mathbb{N}$, it holds that if $\pi_i \in \mathrm{Ps}_{\exists}$ then $\pi_{i+1} = \sigma_{\exists}(\pi_{\leq i})$ else $\pi_{i+1} = \sigma_{\forall}(\pi_{\leq i})$. Intuitively, a play is the unique outcome of the game given by the player strategies. The *play function* $\mathsf{play} : \mathrm{Ps} \times (\mathrm{Str}_{\exists} \times \mathrm{Str}_{\forall}) \rightarrow \mathrm{Pth}$ returns, for each position $v \in \mathrm{Ps}$ and pair of strategies $(\sigma_{\exists}, \sigma_{\forall}) \in \mathrm{Str}_{\exists} \times \mathrm{Str}_{\forall}$, the $((\sigma_{\exists}, \sigma_{\forall}), v)$-play $\mathsf{play}(v, (\sigma_{\exists}, \sigma_{\forall}))$.

2.2. Payoff Arenas

A *payoff arena* is a tuple $\widehat{\mathcal{A}} \triangleq \langle \mathcal{A}, \mathrm{Pf}, \mathsf{pf} \rangle$, where $\mathcal{A}$ is the underlying arena, Pf is the non-empty set of *payoff values*, and $\mathsf{pf} : \mathrm{Pth} \rightarrow \mathrm{Pf}$ is the *payoff function* mapping each path to a value. The *order* of $\widehat{\mathcal{A}}$ is the order of its underlying arena $\mathcal{A}$. A payoff arena is *finite* iff it has finite order. The overloading of the payoff function pf from the set of paths to the sets of positions and pairs of existential and universal strategies induces the function $\mathsf{pf} : \mathrm{Ps} \times (\mathrm{Str}_{\exists} \times \mathrm{Str}_{\forall}) \rightarrow \mathrm{Pf}$ mapping each position $v \in \mathrm{Ps}$ and pair of strategies $(\sigma_{\exists}, \sigma_{\forall}) \in \mathrm{Str}_{\exists} \times \mathrm{Str}_{\forall}$ to the payoff value $\mathsf{pf}(v, (\sigma_{\exists}, \sigma_{\forall})) \triangleq \mathsf{pf}(\mathsf{play}(v, (\sigma_{\exists}, \sigma_{\forall})))$ of the corresponding $((\sigma_{\exists}, \sigma_{\forall}), v)$-play.

2.3. Games

A *(extensive-form) game* is a tuple $\Game \triangleq \langle \widehat{\mathcal{A}}, \mathrm{Wn}, v_{\mathrm{o}} \rangle$, where $\widehat{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Pf}, \mathsf{pf} \rangle$ is the underlying payoff arena, $\mathrm{Wn} \subseteq \mathrm{Pf}$ is the *winning payoff set*, and $v_{\mathrm{o}} \in \mathrm{Ps}$ is the designated *initial position*. The *order* of $\mathcal{G}$ is the order of its underlying payoff arena $\widehat{\mathcal{A}}$. A game is *finite* iff it has finite order. The *existential* (resp., *universal*) *player* $\exists$ (resp., $\forall$) wins the game $\Game$ iff there exists an existential (resp., universal) strategy $\sigma_{\exists} \in \mathrm{Str}_{\exists}$ (resp., $\sigma_{\forall} \in \mathrm{Str}_{\forall}$) such that, for all universal (resp., existential) strategies $\sigma_{\forall} \in \mathrm{Str}_{\forall}$ (resp., $\sigma_{\exists} \in \mathrm{Str}_{\forall}$), it holds that $\mathsf{pf}(\sigma_{\exists}, \sigma_{\forall}) \in \mathrm{Wn}$ (resp., $\mathsf{pf}(\sigma_{\exists}, \sigma_{\forall}) \notin \mathrm{Wn}$). For sake of clarity, given a game $\Game$ we denote with $\mathrm{Pth}(\Game)$ the set of all paths in $\Game$ and with $\mathrm{Str}_{\exists}(\Game)$ and $\mathrm{Str}_{\forall}(\Game)$ the sets of strategies over $\Game$ for the player $\exists$ and $\forall$, respectively. Also, we indicate by $\mathrm{Hst}(\Game)$ the set of the histories over $\Game$.

3. Parity Conditions

In this section, we give an overview about all different parity conditions we consider in this article, which are variants of classical parity games that will be investigated over both classic colored arenas (*i.e.*, with unweighted edges) and weighted arenas. Specifically, along with the known Parity (P), Cost Parity (CP), and Bounded-Cost Parity (BCP) conditions, we introduce three new winning conditions, namely Full Parity (FP), Prompt Parity (PP), and Full-Prompt Parity (FPP).

	Non-Prompt	Prompt
Non-Full	Parity (P)	Prompt Parity (PP) $\equiv$ Cost Parity (CP)
Semi-Full	–	Bounded Cost Parity (BCP)
Full	Full Parity (FP)	Full Prompt Parity (FPP)

Table 1: Prompt/non-prompt conditions under the full/semi-full/non-full constraints.

Before continuing, we introduce some notation to formally define all addressed winning conditions. A *colored arena* is a tuple $\widetilde{\mathcal{A}} \triangleq \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl} \rangle$, where $\mathcal{A}$ is the underlying arena, $\mathrm{Cl} \subseteq \mathbb{N}$ is the non-empty sets of *colors*, and $\mathsf{cl} : \mathrm{Ps} \to \mathrm{Cl}$ is the *coloring function* mapping each position to a color. Similarly, a *(colored) weighted arena* is a tuple $\overline{\mathcal{A}} \triangleq \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$, where $\langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl} \rangle$ is the underlying colored arena, $\mathrm{Wg} \subseteq \mathbb{N}$ is the non-empty sets of *weights*, and $\mathsf{wg} : Mv \to \mathrm{Wg}$ is the *weighting functions* mapping each move to a weight. The overloading of the coloring (resp., weighting) function from the set of positions (resp., moves) to the set of paths induces the function $\mathsf{cl} : \mathrm{Pth} \to \mathrm{Cl}^{\omega}$ (resp., $\mathsf{wg} : \mathrm{Pth} \to \mathrm{Wg}^{\omega}$) mapping each path $\pi \in \mathrm{Pth}$ to the infinite sequence of colors $\mathsf{cl}(\pi) \in \mathrm{Cl}^{\omega}$ (resp. weights $\mathsf{wg}(\pi) \in \mathrm{Wg}^{\omega}$) such that $(\mathsf{cl}(\pi))_i = \mathsf{cl}(\pi_i)$ (resp., $(\mathsf{wg}(\pi))_i = \mathsf{wg}((\pi_i, \pi_{i+1}))$), for all $i \in \mathbb{N}$. Every colored (resp., weighted) arena $\widetilde{\mathcal{A}} \triangleq \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl} \rangle$ (resp., $\overline{\mathcal{A}} \triangleq \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$) induces a canonical payoff arena $\widehat{\mathcal{A}} \triangleq \langle \mathcal{A}, \mathrm{Pf}, \mathsf{pf} \rangle$, where $\mathrm{Pf} \triangleq \mathrm{Cl}^{\omega}$ (resp., $\mathrm{Pf} \triangleq \mathrm{Cl}^{\omega} \times \mathrm{Wg}^{\omega}$) and $\mathsf{pf}(\pi) \triangleq \mathsf{cl}(\pi)$ (resp., $\mathsf{pf}(\pi) \triangleq (\mathsf{cl}(\pi), \mathsf{wg}(\pi))$).

Along a play, we interpret the occurrence of an odd priority as a "*request*" and the occurrence of the first bigger even priority at a later position as a "*response*". Then, we distinguish between *prompt* and *not-prompt* requests. In the not-prompt case, a request is responded independently from the elapsed time between its occurrence and response. Conversely, in the prompt case, the time within a request is responded has an important role. It is for this reason that we consider weighted arenas. So, a *delay* over a play is the sum of the weights over of all the edges crossed from a request to its response. We now formalize these concepts. Let $c \in \mathrm{Cl}^{\omega}$ be an infinite sequence of colors. Then, $\mathrm{Rq}(c) \triangleq \{i \in \mathbb{N} : c_i \equiv 1 \pmod 2\}$ denotes the set of all *requests* in c and $\mathrm{rs}(c, i) \triangleq \min\{j \in \mathbb{N} : i \leq j \wedge c_i \leq c_j \wedge c_j \equiv 0 \pmod 2\}$ represents the *response* to the requests $i \in \mathrm{Rs}$, where by convention we set $\min \emptyset \triangleq \omega$. Moreover, $\mathrm{Rs}(c) \triangleq \{i \in \mathrm{Rq}(c) : \mathrm{rs}(c, i) < \omega\}$ denotes the subset of all requests for which a response is provided. Now, let $w \in \mathrm{Wg}^{\omega}$ be an infinite sequence of weights. Then, $\mathrm{dl}((c, w), i) \triangleq \sum_{k=i}^{\mathrm{rs}(c,i)-1} w_k$ denotes the *delay w.r.t.* w within which a request $i \in \mathrm{Rq}(c)$ is responded. Also, $\mathrm{dl}((c, w), \mathrm{R}) \triangleq \sup_{i \in \mathrm{R}} \mathrm{dl}((c, w), i)$ is the supremum of all delays of the requests contained in $\mathrm{R} \subseteq \mathrm{Rq}(c)$.

As usual, all conditions we consider are given on infinite plays. Then, the winning of the game can be defined *w.r.t.* how often the characterizing properties of the winning condition are satisfied along each play. For example, we may require that *all* requests have to be responded along a play, which we denote as a *full* behavior of the acceptance condition. Also, we may require that the condition (given as a unique or a *conjunction* of properties) holds almost everywhere along the play (*i.e.*, a finite number of places along the play can be ignored), which we denote as a *not-full* behavior of the acceptance condition. More in general, we may have conditions, given as a *conjunction* of several properties, to be satisfied in a mixed way, *i.e.*, some of them have to be satisfied almost everywhere and the remaining ones, over all the play. We denote the latter as a *semi-full* behavior of the acceptance condition. Table 1 reports the combination of the full, not-full, and semi-full behaviors with the known conditions of parity, cost-parity and bounded cost-parity and the new condition of prompt-parity we introduce. As it will be clear in the following, bounded cost-parity has intrinsically a semi-full behavior on weighted arenas, but it has no meaning on (unweighted) colored arenas. Also, over colored arenas, the parity condition has an intrinsic not-full behavior. As far as we known, some of these combinations have never been studied previously on colored arenas (full parity) and weighted arenas (prompt parity and full-prompt parity).

Observe that, in the following, in each graphic representation of a game, the circular nodes belong to player $\exists$ while the square nodes to player $\forall$.

3.1. Non-Prompt Conditions

The non-prompt conditions relate only to the satisfaction of a request (*i.e.*, its response), without taking into account the elapsing of time before the response is provided (*i.e.*, its delay). As reported in Table 1, here we consider as non-prompt conditions, those ones of parity and full parity. To do this, let $\eth \triangleq \langle \widehat{\mathcal{A}}, \mathrm{Wn}, v_0 \rangle$ be a game, where the payoff arena $\widehat{\mathcal{A}}$ is induced by a colored arena $\widetilde{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl} \rangle$.

Parity condition *(*P*)* $\eth$ is a *parity game* iff it is played under a parity condition, which requires that all requests, except at most a finite number, are responded. Formally, for all $c = \mathrm{Cl}^\omega$, we have that $c \in \mathrm{Wn}$ iff there exists a finite set $\mathrm{R} \subseteq \mathrm{Rq}(c)$ such that $\mathrm{Rq}(c) \setminus \mathrm{R} \subseteq \mathrm{Rs}(c)$, *i.e.*, c is a winning payoff iff almost all requests in $\mathrm{Rq}(c)$ are responded. Consider for example the colored arena $\widetilde{\mathcal{A}}_1$ depicted in Figure 1, where all positions are universal, and let $\alpha + \beta$ be the regular expression describing all possible plays starting at v_0, where $\alpha = (v_0 \cdot v_1^* \cdot v_2) \cdot v_0 \cdot v_1^\omega$ and $\beta = (v_0 \cdot v_1^* \cdot v_2)^\omega$. Now, keep a path $\pi \in \alpha$ and let $c_\alpha \triangleq \mathsf{pf}(\pi) \in (1 \cdot 0^* \cdot 2) \cdot 1 \cdot 0^\omega$ be its payoff. Then, $c_\pi \in \mathrm{Wn}$, since the parity condition is satisfied by putting in R the last index in which the color 1 occurs in c_π. Again, keep a path $\pi \in \beta$ and let $c_\pi \triangleq \mathsf{pf}(\pi) \in (1 \cdot 0^* \cdot 2)^\omega$ be its payoff. Then, $c_\pi \in \mathrm{Wn}$, since the parity condition is satisfied by simply choosing $\mathrm{R} \triangleq \emptyset$. In the following, as a special case, we also consider parity games played over arenas colored only with the two priorities 1 and 2, to which we refer as *Büchi games* (B).

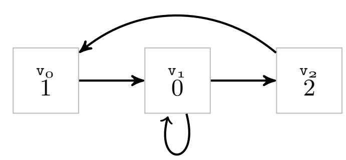

Figure 1: Colored Arena $\widetilde{\mathcal{A}}_1$.

Full Parity condition *(*FP*)* $\eth$ is a *full parity game* iff it is played under a full parity condition, which requires that all requests are responded. Formally, for all $c \in \mathrm{Cl}^\omega$, we have that $c \in \mathrm{Wn}$ iff $\mathrm{Rq}(c) \subseteq \mathrm{Rs}(c)$ *i.e.*, c is a winning payoff iff all requests in $\mathrm{Rq}(c)$ are responded. Consider for example the colored arena $\widetilde{\mathcal{A}}_2$ in Figure 2, where all positions are existential. There is a unique path $\pi = (v_0 \cdot v_1)^\omega$ starting at v_0 having payoff $c_\pi \triangleq \mathsf{pf}(\pi) = (1 \cdot 2)^\omega$ and set of requests $\mathrm{Rq}(c_\pi) = \{2n : n \in \mathbb{N}\}$. Then, $c_\pi \in \mathrm{Wn}$, since the full parity condition is satisfied as all requests are responded by the color 2 at the odd indexes. Observe that the arena of the game $\widetilde{\mathcal{A}}_1$ depicted in Figure 1 is not won under the *full parity* condition. Indeed, if we consider the path π with payoff $\mathsf{pf}(\pi) \in (1 \cdot 0^\omega)$, it holds that not all requests are responded.

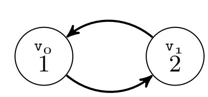

Figure 2: Colored Arena $\widetilde{\mathcal{A}}_2$.

3.2. Prompt Conditions

The prompt conditions take into account, in addition to the satisfaction of a request, also the delay before it occurs. As reported in Table 1, here we consider as prompt conditions, those ones of prompt parity, full-prompt parity, cost parity, and bounded-cost parity. To do this, let $\Game \triangleq \langle \widehat{\mathcal{A}}, \mathrm{Wn}, v_{0} \rangle$ be a game, where the payoff arena $\widehat{\mathcal{A}}$ is induced by a (colored) weighted arena $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$.

Prompt Parity condition *(*PP*)* $\Game$ is a *prompt parity game* iff it is played under a prompt parity condition, which requires that all requests, except at most a finite number of them, are responded with a bounded delay. Formally, for all $(c, w) \in \mathrm{Cl}^{\omega} \times \mathrm{Wg}^{\omega}$, we have that $(c, w) \in \mathrm{Wn}$ iff there exists a finite set $\mathrm{R} \subseteq \mathrm{Rq}(c)$ such that $\mathrm{Rq}(c) \setminus \mathrm{R} \subseteq \mathrm{Rs}(c)$ and there exists a bound $b \in \mathbb{N}$ for which $\mathrm{dl}((c, w), \mathrm{Rq}(c) \setminus \mathrm{R}) \leq b$ holds, *i.e.*, (c, w) is a winning payoff iff almost all requests in $\mathrm{Rq}(c)$ are responded with a delay bounded by an a priori number b. Consider for example the weighted arena $\overline{\mathcal{A}_3}$ depicted in Figure 3. There is a unique path $\pi = v_0 \cdot (v_1 \cdot v_2)^{\omega}$ starting at v_0 having payoff $p_{\pi} \triangleq \mathsf{pf}(\pi) = (c_{\pi}, w_{\pi})$, where $c_{\pi} = 3 \cdot (1 \cdot 2)^{\omega}$ and $w_{\pi} = 2 \cdot (1 \cdot 0)^{\omega}$, and set of requests $\mathrm{Rq}(c_{\pi}) = \{0\} \cup \{2n + 1 : n \in \mathrm{N}\}$. Then, $p_{\pi} \in \mathrm{Wn}$, since the prompt parity condition is satisfied by choosing $\mathrm{R} = \{0\}$ and $b = 1$.

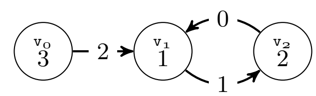

Figure 3: Weighted Arena $\overline{\mathcal{A}_3}$.

Full-Prompt Parity condition *(*FPP*)* $\Game$ is a *full-prompt parity game* iff it is played under a full-prompt parity condition, which requires that all requests are responded with a bounded delay. Formally, for all $(c, w) \in \mathrm{Cl}^{\omega} \times \mathrm{Wg}^{\omega}$, we have that $(c, w) \in \mathrm{Wn}$ iff $\mathrm{Rq}(c) = \mathrm{Rs}(c)$ and there exists a bound $b \in \mathbb{N}$ for which $\mathrm{dl}((c, w), \mathrm{Rq}(c)) \leq b$ holds, *i.e.*, (c, w) is a winning payoff iff all requests in $\mathrm{Rq}(c)$ are responded with a delay bounded by an a priori number b. Consider for example the weighted arena $\overline{\mathcal{A}_4}$ depicted in Figure 4. Now, take a path $\pi \in v_0 \cdot v_1 \cdot ((v_0 \cdot v_1)^* \cdot (v_2 \cdot v_1)^*)^{\omega}$ starting at v_0 and let $p_{\pi} \triangleq \mathsf{pf}(\pi) = (c_{\pi}, w_{\pi})$ be its payoff, with $c_{\pi} \in 3 \cdot 4 \cdot ((3 \cdot 4)^* \cdot (1 \cdot 4)^*)^{\omega}$ and $w_{\pi} \in 2 \cdot ((0 \cdot 2)^* \cdot (0 \cdot 1)^*)^{\omega}$. Then, $p_{\pi} \in \mathrm{Wn}$, since the full-prompt parity condition is satisfied as all requests are responded by color 4 with a delay bound $b = 2$. Observe that, the arena of the game $\widetilde{\mathcal{A}_3}$ depicted in Figure 3 is not won under the *full prompt parity* condition. Indeed, if we consider the unique path $\pi = v_0 \cdot (v_1 \cdot v_2)^{\omega}$ starting at v_0 having payoff $p_{\pi} \triangleq \mathsf{pf}(\pi) = (c_{\pi}, w_{\pi})$, where $c_{\pi} = 3 \cdot (1 \cdot 2)^{\omega}$ and $w_{\pi} = 2 \cdot (1 \cdot 0)^{\omega}$, it holds that there exists an unanswered request at the vertex v_0.

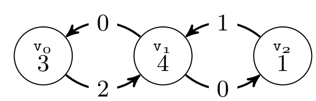

Figure 4: Weighted Arena $\overline{\mathcal{A}_4}$.

Remark 3.1. As a special case, the prompt and the full-prompt parity conditions can be analyzed on simply colored arenas, by considering each edge as having weight 1. Then, the two cases just analyzed correspond to the finitary parity and bounded parity conditions studied in [11].

Cost Parity condition *(*CP*) [17]* $\Game$ is a *cost parity game* iff it is played under a cost parity condition, which requires that all requests, except at most a finite number of them, are responded and all requests, except at most a finite number of them (possibly different from the previous ones) have a bounded delay. Formally,

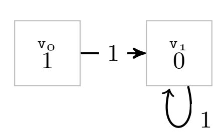

Figure 5: Weighted Arena $\overline{\mathcal{A}_5}$.

for all $(c, w) \in \mathrm{Cl}^{\omega} \times \mathrm{Wg}^{\omega}$, we have that $(c, w) \in \mathrm{Wn}$ iff there is a finite set $\mathrm{R} \subseteq \mathrm{Rq}(c)$ such that $\mathrm{Rq}(c) \setminus \mathrm{R} \subseteq \mathrm{Rs}(c)$ and there exist a finite set $\mathrm{R}' \subseteq \mathrm{Rq}(c)$ and a bound $b \in \mathbb{N}$ for which $\mathrm{dl}((c, w), \mathrm{Rq}(c) \setminus \mathrm{R}') \leq b$ holds, *i.e.*, (c, w) is a winning payoff iff almost all requests in $\mathrm{Rq}(c)$ are responded and almost all have a delay bounded by an a priori number b. Consider for example the weighted arena $\overline{\mathcal{A}_5}$ in Figure 5. There is a unique path $\pi = v_0 \cdot {v_1}^{\omega}$ starting at v_0 having payoff $p_\pi \triangleq \mathsf{pf}(\pi) = (c_\pi, w_\pi)$, where $c_\pi = 1 \cdot 0^{\omega}$ and $w_\pi = 1^{\omega}$, and set of requests $\mathrm{Rq}(c_\pi) = \{0\}$. Then, $p_\pi \in \mathrm{Wn}$, since the prompt parity condition is satisfied with $\mathrm{R} = \mathrm{R}' = \{0\}$ and $b = 0$.

Bounded-Cost Parity condition *(*BCP*) [17]* $\Game$ is a *bounded-cost parity game* iff it is played under a bounded-cost parity condition, which requires that all requests, except at most a finite number, are responded and all have a bounded delay. Formally, for all $(c, w) \in \mathrm{Cl}^{\omega} \times \mathrm{Wg}^{\omega}$, we have that $(c, w) \in \mathrm{Wn}$ iff there exists a finite set $\mathrm{R} \subseteq \mathrm{Rq}(c)$ such that $\mathrm{Rq}(c) \setminus \mathrm{R} \subseteq \mathrm{Rs}(c)$ and there exists a bound $b \in \mathbb{N}$ for which $\mathrm{dl}((c, w), \mathrm{Rq}(c)) \leq b$ holds, *i.e.*, (c, w) is a winning payoff iff almost all requests in $\mathrm{Rq}(c)$ are responded and all have a delay bounded by an a priori number b. Consider for example the weighted arena $\overline{\mathcal{A}_6}$ depicted in Figure 6. There is a unique path $\pi = v_0 \cdot {v_1}^{\omega}$ starting at v_0 having payoff $p_\pi \triangleq \mathsf{pf}(\pi) = (c_\pi, w_\pi)$, where $c_\pi = 1 \cdot 0^{\omega}$, and set of requests $\mathrm{Rq}(c_\pi) = \{0\}$. Then, $p_\pi \in \mathrm{Wn}$, since the prompt parity condition is satisfied with $\mathrm{R} = \{0\}$ and $b = 1$.

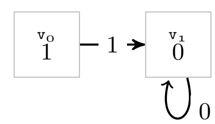

Figure 6: Weighted Arena $\overline{\mathcal{A}_6}$.

Wn	Formal definitions		
P	$\forall c \in \mathrm{Cl}^{\omega}.\ c \in \mathrm{Wn}$ iff	$\exists \mathrm{R} \subseteq \mathrm{Rq}(c), \lvert\mathrm{R}\rvert < \omega.$	$\mathrm{Rq}(c) \setminus \mathrm{R} \subseteq \mathrm{Rs}(c)$
FP			$\mathrm{Rq}(c) = \mathrm{Rs}(c)$
PP	$\forall (c, w) \in \mathrm{Cl}^{\omega} \times \mathrm{Wg}^{\omega}.$ $(c, w) \in \mathrm{Wn}$ iff	$\exists \mathrm{R} \subseteq \mathrm{Rq}(c), \lvert\mathrm{R}\rvert < \omega.$	$\mathrm{Rq}(c) \setminus \mathrm{R} \subseteq \mathrm{Rs}(c) \wedge$ $\exists b \in \mathbb{N}\ .\ \mathrm{dl}((c, w), \mathrm{Rq}(c) \setminus \mathrm{R}) \leq b$
FPP			$\mathrm{Rq}(c) = \mathrm{Rs}(c) \wedge$ $\exists b \in \mathbb{N}\ .\ \mathrm{dl}((c, w), \mathrm{Rq}(c)) \leq b$
CP		$\exists \mathrm{R} \subseteq \mathrm{Rq}(c), \lvert\mathrm{R}\rvert < \omega.$ $\exists \mathrm{R}' \subseteq \mathrm{Rq}(c), \lvert\mathrm{R}'\rvert < \omega.$	$\mathrm{Rq}(c) \setminus \mathrm{R} \subseteq \mathrm{Rs}(c) \wedge$ $\exists b \in \mathbb{N}\ .\ \mathrm{dl}((c, w), \mathrm{Rq}(c) \setminus \mathrm{R}') \leq b$
BCP		$\exists \mathrm{R} \subseteq \mathrm{Rq}(c), \lvert\mathrm{R}\rvert < \omega.$	$\mathrm{Rq}(c) \setminus \mathrm{R} \subseteq \mathrm{Rs}(c) \wedge$ $\exists b \in \mathbb{N}\ .\ \mathrm{dl}((c, w), \mathrm{Rq}(c)) \leq b$

Table 2: Summary of all winning condition (Wn) definitions.

In Table 2, we list all winning conditions (Wn) introduced above, along with their respective formal definitions. For the sake of readability, given a game $\Game = \langle \widehat{\mathcal{A}}, \mathrm{Wn}, v_0 \rangle$, we sometimes use the winning condition acronym name in place of Wn, as well as we refer to $\Game$ as a Wn game. For example, if $\Game$ is a parity game, we also say that it is a P game, as well as, write $\Game = \langle \widehat{\mathcal{A}}, \mathrm{P}, v_0 \rangle$.

4. Equivalences and Implications

In this section, we investigate the relationships among all parity conditions discussed above.

4.1. Positive Relationships

In this subsection, we prove all positive existing relationships among the studied conditions and report them in Figure 7, where an arrow from a condition Wn_1 to another condition Wn_2 means that the former implies the latter. In other words, if player $\exists$ wins a game under the condition Wn_1, then it also wins the game under the condition Wn_2, over the same arena. The label on the edges indicates the item of the next theorem in which the result is proved. In particular, we show that prompt parity and cost parity are semantically equivalent. The same holds for full parity and full prompt parity over finite arenas and for full prompt parity and bounded cost parity on positive weighted arenas. Also, as one may expect, fullness implies not-fullness under every condition and all conditions imply the parity one.

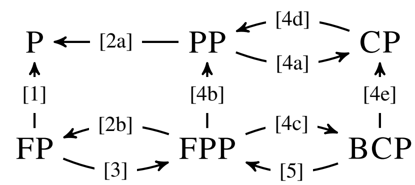

Figure 7: Implication Schema.

Theorem 4.1. Let $\Game_1 = \langle \widehat{\mathcal{A}}_1, \mathrm{Wn}_1, v_0 \rangle$ and $\Game_2 = \langle \widehat{\mathcal{A}}_2, \mathrm{Wn}_2, v_0 \rangle$ be two games defined on the payoff arenas $\widehat{\mathcal{A}}_1$ and $\widehat{\mathcal{A}}_2$ having the same underlying arena $\mathcal{A}$. Then, player $\exists$ wins $\Game_2$ if it wins $\Game_1$ under the following constraints:

1. $\widehat{\mathcal{A}}_1 = \widehat{\mathcal{A}}_2$ are induced by a colored arena $\widetilde{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl} \rangle$ and $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{FP}, \mathrm{P})$;

2. $\widehat{\mathcal{A}}_1$ and $\widehat{\mathcal{A}}_2$ are induced, respectively, by a weighted arena $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$ and its underlying colored arena $\widetilde{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl} \rangle$ and

   (a) $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{PP}, \mathrm{P})$, or

   (b) $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{FPP}, \mathrm{FP})$;

3. $\widehat{\mathcal{A}}_2$ and $\widehat{\mathcal{A}}_1$ are finite and induced, respectively, by a weighted arena $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$ and its underlying colored arena $\widetilde{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl} \rangle$ and $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{FP}, \mathrm{FPP})$;

4. $\widehat{\mathcal{A}}_1 = \widehat{\mathcal{A}}_2$ are induced by a weighted arena $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$ and

   (a) $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{PP}, \mathrm{CP})$, or

   (b) $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{FPP}, \mathrm{PP})$, or

   (c) $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{FPP}, \mathrm{BCP})$, or

   (d) $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{CP}, \mathrm{PP})$, or

   (e) $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{BCP}, \mathrm{CP})$;

5. $\widehat{\mathcal{A}}_1 = \widehat{\mathcal{A}}_2$ are induced by a weighted arena $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$, with $\mathsf{wg}(v) > 0$ for all $v \in \mathrm{Ps}$, and $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{BCP}, \mathrm{FPP})$.

Proof:
All items, but 3, 4d, and 5, are immediate by definition. So, we only focus on the remaining ones.

[Item 3] Suppose by contradiction that player $\exists$ wins the FP $\Game_1$ game but it does not win the FPP game $\Game_2$. Then, there is a play π in $\Game_2$ having payoff $(c, w) = \mathsf{pf}(\pi) \in \mathrm{Cl}^\omega \times \mathrm{Wg}^\omega$ for which $\mathrm{dl}((c, w), \mathrm{Rq}(c)) = \omega$. So, there exists at least a request $r \in \mathrm{Rq}(c)$ with a delay greater than $s = \sum_{e \in Mv} \mathsf{wg}(e)$. Since the arena is finite, this implies that, on the infix of π that goes from the request r to its response, there is a move that occurs twice. So, player $\forall$ has the possibility to force another play π' having r as request and passing infinitely often through this move without reaching the response. But this is impossible, since player $\exists$ wins the FP game $\Game_1$.

[Item 4d] To prove this item, we show that if a payoff $(c, w) \in \mathrm{Cl}^\omega \times \mathrm{Wg}^\omega$ satisfies the CP condition then it also satisfies the PP one. Indeed, by definition, there are a finite set $\mathrm{R} \subseteq \mathrm{Rq}(c)$ such that $\mathrm{Rq}(c) \setminus \mathrm{R} \subseteq \mathrm{Rs}(c)$ and a possibly different finite set $\mathrm{R}^{'} \subseteq \mathrm{Rq}(c)$ for which there is a bound $b \in \mathbb{N}$ such that $\mathrm{dl}((c, w), \mathrm{Rq}(c) \setminus \mathrm{R}^{'}) \leq b$. Now, consider the union $\mathrm{R}^{''} \triangleq \mathrm{R} \cup \mathrm{R}^{'}$. Obviously, this is a finite set. Moreover, it is immediate to see that $\mathrm{Rq}(c) \setminus \mathrm{R}'' \subseteq \mathrm{Rs}(c)$ and $\mathrm{dl}((c, w), \mathrm{Rq}(c) \setminus \mathrm{R}^{''}) \leq b$, for the same bound b. So, the payoff (c, w) satisfies the PP condition, by using $\mathrm{R}^{''}$ in place of R in the definition.

[Item 5] Suppose by contradiction that player $\exists$ wins the BCP game $\Game_1$ but it does not win the FPP game $\Game_2$. Then, there is a play π in $\Game_2$ having payoff $(c, w) = \mathsf{pf}(\pi) \in \mathrm{Cl}^\omega \times \mathrm{Wg}^\omega$ for which $\mathrm{Rq}(c) \neq \mathrm{Rs}(c)$. So, since all weights are positive, there exists at least a request $r \in \mathrm{Rq}(c) \setminus \mathrm{Rs}(c) \neq \emptyset$ with $\mathrm{dl}((c, w), r) = \omega$. But this is impossible. □

The following three corollaries follow as immediate consequences of, respectively, Items 2b and 3, 4a and 4d, and 4c and 5 of the previous theorem.

Corollary 4.2. Let $\Game_{\mathrm{FPP}} = \langle \widehat{\mathcal{A}_{\mathrm{FPP}}}, \mathrm{FPP}, v_0 \rangle$ be an FPP game and $\Game_{\mathrm{FP}} = \langle \widehat{\mathcal{A}_{\mathrm{FP}}}, \mathrm{FP}, v_0 \rangle$ an FP one defined on the two finite payoff arenas $\widehat{\mathcal{A}_{\mathrm{FPP}}}$ and $\widehat{\mathcal{A}_{\mathrm{FP}}}$ induced, respectively, by a weighted arena $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$ and its underlying colored arena $\widetilde{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl} \rangle$. Then, player $\exists$ wins $\Game_{\mathrm{FPP}}$ if it wins $\Game_{\mathrm{FP}}$.

Corollary 4.3. Let $\Game_{\mathrm{CP}}=\langle \widehat{\mathcal{A}}, \mathrm{CP}, v_0 \rangle$ be a CP game and $\Game_{\mathrm{PP}}=\langle \widehat{\mathcal{A}}, \mathrm{PP}, v_0 \rangle$ a PP one defined on the payoff arena $\widehat{\mathcal{A}}$ induced by a weighted arena $\overline{\mathcal{A}}=\langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$. Then, player $\exists$ wins $\Game_{\mathrm{CP}}$ if it wins $\Game_{\mathrm{PP}}$.

Corollary 4.4. Let $\Game_{\mathrm{BCP}} = \langle \widehat{\mathcal{A}}, \mathrm{BCP}, v_0 \rangle$ be a BCP game and $\Game_{\mathrm{FPP}} = \langle \widehat{\mathcal{A}}, \mathrm{FPP}, v_0 \rangle$ an FPP one defined on the payoff arena $\widehat{\mathcal{A}}$ induced by a weighted arena $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$, where $\mathsf{wg}(v) > 0$, for all $v \in \mathrm{Ps}$. Then, player $\exists$ wins $\Game_{\mathrm{BCP}}$ if it wins $\Game_{\mathrm{FPP}}$.

4.2. Negative Relationships

In this subsection, we show a list of counterexamples to point out that some winning conditions are not equivalent to other ones. We report the corresponding result in Figure 8, where an arrow from a condition Wn_1 to another condition Wn_2 means that there exists an arena on which player $\exists$ wins a Wn_1 game while it loses a Wn_2 one. The label on the edges indicates the item of

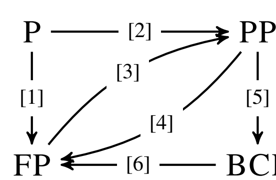

Figure 8: Counterexample Schema.

the next theorem in which the result is proved. Moreover, the following list of counter-implications, non reported in the figure, can be simply obtained by the reported ones together with the implication results of Theorem 4.1: (P, FPP), (P, CP), (P, BCP), (FP, FPP), (FP, CP), (FP, BCP), (PP, FPP), (CP, FP), (CP, FPP), (CP, BCP), and (BCP, FPP).

Theorem 4.5. There exist two games $\Game_1 = \langle \widehat{\mathcal{A}_1}, \mathrm{Wn}_1, v_0 \rangle$ and $\Game_2 = \langle \widehat{\mathcal{A}_2}, \mathrm{Wn}_2, v_0 \rangle$, defined on the two payoff arenas $\widehat{\mathcal{A}_1}$ and $\widehat{\mathcal{A}_2}$ having the same underlying arena $\mathcal{A}$, such that player $\exists$ wins $\Game_1$ while it loses $\Game_2$ under the following constraints:

1. $\widehat{\mathcal{A}_1} = \widehat{\mathcal{A}_2}$ are induced by a colored arena $\widetilde{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl} \rangle$ and $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{P}, \mathrm{FP})$;

2. $\widehat{\mathcal{A}_2}$ and $\widehat{\mathcal{A}_1}$ are induced, respectively, by a weighted arena $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$ and its underlying colored arena $\widetilde{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl} \rangle$ and $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{P}, \mathrm{PP})$;

3. $\widehat{\mathcal{A}_2}$ and $\widehat{\mathcal{A}_1}$ are infinite and induced, respectively, by a weighted arena $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$ and its underlying colored arena $\widetilde{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl} \rangle$ and $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{FP}, \mathrm{PP})$;

4. $\widehat{\mathcal{A}_1}$ and $\widehat{\mathcal{A}_2}$ are induced, respectively, by a weighted arena $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$ and its underlying colored arena $\widetilde{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl} \rangle$ and $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{PP}, \mathrm{FP})$;

5. $\widehat{\mathcal{A}_1} = \widehat{\mathcal{A}_2}$ are induced by a weighted arena $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$ and $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{PP}, \mathrm{BCP})$;

6. $\widehat{\mathcal{A}_1}$ and $\widehat{\mathcal{A}_2}$ are induced, respectively, by a weighted arena $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$, with $\mathsf{wg}(v) = 0$ for some $v \in \mathrm{Ps}$, and its underlying colored arena $\widetilde{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl} \rangle$ and $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{BCP}, \mathrm{FP})$.

Proof:
[Item 1] Consider as colored arena $\widetilde{\mathcal{A}}$ the one underlying the weighted arena depicted in Figure 5, which has just the path $\pi = v_0 \cdot {v_1}^\omega$ with payoff $c = \mathsf{pf}(\pi) = 1 \cdot 0^\omega$. Then, it is immediate to see that player $\exists$ wins the P game but not the FP game, since $\mathrm{Rq}(c) \setminus \mathrm{Rs}(c) = \{0\}$.

[Item 2] Consider as colored arena $\widetilde{\mathcal{A}}$ the one depicted in Figure 1 and as weighted arena $\overline{\mathcal{A}}$ the one having a weight 1 on the self loop on v_1 and 0 on the remaining moves. It is immediate to see that player $\exists$ wins the P game $\Game_1$. However, player $\forall$ has a strategy that forces in the PP game $\Game_2$ the play $\pi = \prod_{i=1}^{\omega} v_0 \cdot {v_1}^i \cdot v_2$ having payoff $(c, w) = \mathsf{pf}(\pi) = (\prod_{i=1}^{\omega} 1 \cdot 0^i \cdot 2, \prod_{i=1}^{\omega} 0 \cdot 1^i \cdot 0)$. Therefore, player $\exists$ does not win $\Game_2$, since there is no finite set $\mathrm{R} \subset \mathrm{Rq}(c)$ for which $\mathrm{dl}((c, w), \mathrm{Rq}(c) \setminus \mathrm{R}) < \omega$.

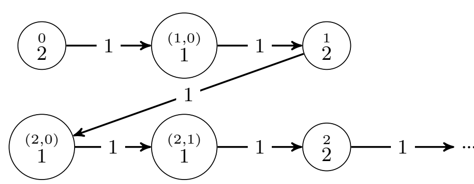

Figure 9: Infinite Weighted Arena $\overline{\mathcal{A}_7}$.

[Item 3] Consider as weighted arena $\overline{\mathcal{A}}$ the infinite one depicted in Figure 9 having set of positions $\mathrm{Ps} \triangleq \mathbb{N} \cup \{(i, j) \in \mathbb{N} \times \mathbb{N} : j < i\}$ and moves defined as follows: if $j < i - 1$ then $((i, j), (i, j+1)) \in \mathit{Mv}$ else $((i, j), i) \in \mathit{Mv}$. In addition, the coloring of the positions are $\mathsf{cl}(i) = 2$ and $\mathsf{cl}((i, j)) = 1$. Now, it is immediate to see that, on the underlying colored arena $\widetilde{\mathcal{A}}$, Player $\exists$ wins the FP game $\Game_1$, since all

requests on the unique possible play $\pi = \prod_{i=0}^{\omega}(\prod_{j=0}^{i-1}(i,j)) \cdot i$ are responded. However, it does not win the PP game $\Game_2$, since $\mathrm{dl}((c,w), \mathrm{Rq}(c)) = \omega$, where $(c,w) = \mathsf{pf}(\pi) = (\prod_{i=0}^{\omega} 1^i \cdot 2, 1^\omega)$. Indeed, there is no finite set $\mathrm{R} \subset \mathrm{Rq}(c)$ for which $\mathrm{dl}((c,w), \mathrm{Rq}(c) \setminus \mathrm{R}) < \omega$.

[Item 4] Consider as weighted arena $\overline{\mathcal{A}}$ the one depicted in Figure 5 having just the path $\pi = v_0 \cdot {v_1}^\omega$ with payoff $(c,w) = \mathsf{pf}(\pi) = (1 \cdot 0^\omega, 0 \cdot 1^\omega)$. Player $\exists$ wins the PP game $\Game_1$, since there is just one requests, which we can simply avoid to consider. However, as already observed in Item 1, the FP game $\Game_2$ on the underlying colored arena $\widetilde{\mathcal{A}}$ is not won by the same player.

[Item 5] Consider again as weighted arena $\overline{\mathcal{A}}$ the one depicted in Figure 5. As already observed in Item 4, the PP game $\Game_1$ is won by player $\exists$. However, it does not win the BCP game $\Game_2$, since $\mathrm{dl}((c,w), 0) = \omega$.

[Item 6] Consider as weighted arena $\overline{\mathcal{A}}$ the one depicted in Figure 6 having just the path $\pi = v_0 \cdot {v_1}^\omega$ with payoff $(c,w) = \mathsf{pf}(\pi) = (1 \cdot 0^\omega, 1 \cdot 0^\omega)$. Player $\exists$ wins the BCP game $\Game_1$, since there is just one requests, which we can simply avoid to consider, and its delay is equal to 1. However, as already observed in Item 1, the FP game $\Game_2$ on the underlying colored arena $\widetilde{\mathcal{A}}$ is not won by the same player. □

5. Polynomial Reductions

In this section, we face the computational complexity of solving full parity, prompt parity and bounded cost parity games. Then, due to the relationships among the winning conditions described in the previous section, we propagate the achieved results to the other conditions as well. The technique we adopt is to solve a given game through the construction of a new game over an enriched arena, on which we play with a simpler winning condition. Intuitively, the constructed game encapsulates, inside the states of its arena, some information regarding the satisfaction of the original condition. To this aim, we introduce the concepts of *transition table* and its *product* with an arena. Intuitively, a transition table is an automaton without acceptance condition, which is used to represent the information of the winning condition mentioned above. Then, the product operation allows to inject this information into the new arena. In general, our constructions are pseudo-polynomial, but if we restrict to the case of having only 0 and 1 as weights over the edges, then they become polynomial, due to the fact that the threshold is bounded by the number of edges in the arena. Moreover, since a game with arbitrary weights can be easily transformed into one with weights 0 and 1, we overall get a polynomial reduction for all the cases. Note that to check whether a value is positive or zero can be done in linear time in the number of its bits and, therefore, it is linear in the description of its weights.

In the following, for a given set of colors $\mathrm{Cl} \subseteq \mathbb{N}$, we assume $\bot < i$, for all $i \in \mathrm{Cl}$. Intuitively, $\bot$ is a special symbol that can be seen as lower bound over color priorities. Moreover, we define $\mathrm{R} \triangleq \{c \in \mathrm{Cl} : c \equiv 1 \pmod{2}\}$ to be the set of all possible request values in Cl with $\mathrm{R}_\bot \triangleq \{\bot\} \cup \mathrm{R}$.

5.1. Transition Tables

A *transition table* is a tuple $\mathcal{T} \triangleq \langle \mathrm{Sm}, \mathrm{St}_D, \mathrm{St}_\exists, \mathsf{tr} \rangle$, where Sm is the set of *symbols*, St_D and $\mathrm{St}_\exists$ with $\mathrm{St} \triangleq \mathrm{St}_D \cup \mathrm{St}_\exists$ are disjoint sets of *deterministic* and *existential states*, and $\mathsf{tr} : (\mathrm{St}_D \times \mathrm{Sm} \to \mathrm{St}) \cup (\mathrm{St}_\exists \to 2^{\mathrm{St}})$ is the *transition function* mapping either pairs of deterministic states and symbols to states or existential states to sets of states. The *order* (resp., *size*) of $\mathcal{T}$ is $|\mathcal{T}| \triangleq |\mathrm{St}|$ (resp., $\|\mathcal{T}\| \triangleq |\mathsf{tr}|$). A transition table is *finite* iff it has finite order.

Let $\widetilde{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl} \rangle$ be a colored arena with $\mathcal{A} = \langle \mathrm{Ps}_\exists, \mathrm{Ps}_\forall, Mv \rangle$ and $\mathcal{T} \triangleq \langle \mathrm{Cl}, \mathrm{St}_D, \mathrm{St}_\exists, \mathsf{tr} \rangle$ a transition table. Then, $\widetilde{\mathrm{Ar}} \otimes \mathcal{T} \triangleq \langle \mathrm{Ps}^\star_\exists, \mathrm{Ps}^\star_\forall, Mv^\star \rangle$ is the *product arena* defined as follows:

- $\mathrm{Ps}^\star_\exists \triangleq \mathrm{Ps}_\exists \times \mathrm{St}_D \cup \mathrm{Ps} \times \mathrm{St}_\exists$ and $\mathrm{Ps}^\star_\forall \triangleq \mathrm{Ps}_\forall \times \mathrm{St}_D$;
- for $(v_1, v_2) \in Mv$ and $s \in \mathrm{St}_D$, it holds that $((v_1, s), (v_2, \mathsf{tr}(s, \mathsf{cl}(v_1)))) \in Mv^\star$;
- for $v \in \mathrm{Ps}$, $s_1 \in \mathrm{St}_\exists$, and $s_2 \in \mathrm{St}$, then, $((v, s_1), (v, s_2)) \in Mv^\star$ iff $s_2 \in \mathsf{tr}(s_1)$.

Similarly, let $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$ be a weighted arena with $\mathcal{A} = \langle \mathrm{Ps}_\exists, \mathrm{Ps}_\forall, Mv \rangle$ and $\mathcal{T} \triangleq \langle \mathrm{Cl} \times \mathrm{Wg}, \mathrm{St}_D, \mathrm{St}_\exists, \mathsf{tr} \rangle$ a transition table. Then, $\overline{\mathrm{Ar}} \otimes \mathcal{T} \triangleq \langle \mathrm{Ps}^\star_\exists, \mathrm{Ps}^\star_\forall, Mv^\star \rangle$ is the *product arena* as before, except for all moves $(v_1, v_2) \in Mv$ and states $s \in \mathrm{St}_D$, where we have that $((v_1, s), (v_2, \mathsf{tr}(s, (\mathsf{cl}(v_1), \mathsf{wg}((v_1, v_2)))))) \in Mv^\star$.

5.2. From Full Parity to Büchi

In this section, we show a reduction from full parity games to Büchi ones. The reduction is done by constructing an ad-hoc transition table $\mathcal{T}$ that maintains basic informations of the parity condition. Then, the Büchi game uses as an arena an enriched version of the original one, which is obtained as its product with the built transition table. Intuitively, the latter keeps track, along every play, of the value of the biggest unanswered request. When such a request is satisfied, this value is set to the special symbol $\perp$. To this aim, we use as states of the transition table, together with the symbol $\perp$, all possible request values. Also, the transition function is defined in the following way: if a request is satisfied then it moves to state $\perp$, otherwise, it moves to the state representing the maximum between the new request it reads and the previous memorized one (kept into the current state). Consider now the arena built as the product of the original arena with the above described transition table and use as colors the values 1 and 2, assigned as follows: if a position contains $\perp$, color it with 2, otherwise, with 1. By definition of full parity and Büchi games, we have that a Büchi game is won over the new built arena if and only if the the full parity game is won over the original arena. Indeed, over a play of the new arena, meeting a bottom symbol infinitely often means that all requests found over the corresponding play of the old arena are satisfied. The formal construction of the transition table and the enriched arena follow. For a given FP game $\Game \triangleq \langle \widehat{\mathcal{A}}, \mathrm{FP}, v_\mathrm{o} \rangle$ induced by a colored arena $\widetilde{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl} \rangle$, we construct a deterministic transition table $\mathcal{T} \triangleq \langle \mathrm{Cl}, \mathrm{St}, \mathsf{tr} \rangle$, with set of states $\mathrm{St} \triangleq \mathrm{R}_\perp$ and transition function defined as follows:

- $\mathsf{tr}(r, c) \triangleq \begin{cases} \perp, & \text{if } r < c \text{ and } c \equiv 0 \pmod 2; \\ \max\{r, c\}, & \text{otherwise.} \end{cases}$

Now, let $\mathcal{A}^\star = \widetilde{\mathcal{A}} \otimes \mathcal{T}$ be the product arena of $\widetilde{\mathcal{A}}$ and $\mathcal{T}$ and consider the colored arena $\widetilde{\mathcal{A}^\star} \triangleq \langle \mathcal{A}^\star, \{1, 2\}, \mathsf{cl}^\star \rangle$ such that, for all positions $(v, r) \in \mathrm{Ps}^\star$, if $r = \perp$ then $\mathsf{cl}^\star((v, r)) = 2$ else $\mathsf{cl}^\star((v, r)) = 1$. Then, the B game $\Game^\star = \langle \widehat{\mathcal{A}^\star}, \mathrm{B}, (v_\mathrm{o}, \perp) \rangle$ induced by $\widetilde{\mathcal{A}^\star}$ is such that player $\exists$ wins $\Game$ iff it wins $\Game^\star$.

Theorem 5.1. For every FP game $\Game$ with $k \in \mathbb{N}$ priorities, there is a B game $\Game^\star$, with order $|\Game^\star| = \mathrm{O}(|\Game| \cdot k)$, such that player $\exists$ wins $\Game$ iff it wins $\Game^\star$.

Proof:
[If] By hypothesis, we have that player $\exists$ wins the B game $\Game^\star$ on the colored arena $\widetilde{\mathcal{A}}$, which induces a

payoff arena $\widehat{\mathcal{A}}$. This means that, there exists a strategy $\sigma^{\star}_{\exists} \in \mathrm{Str}_{\exists}(\Game^{*})$ for player $\exists$ such that for each strategy $\sigma^{\star}_{\forall} \in \mathrm{Str}_{\forall}(\Game^{*})$ for player $\forall$, it holds that $\mathsf{pf}(v_{0}, (\sigma^{\star}_{\exists}, \sigma^{\star}_{\forall})) \in \mathrm{B}$. Therefore, for all $\pi^{\star} \in \mathrm{Pth}(\Game^{\star}_{\restriction \sigma^{\star}_{\exists}})$, we have that $\mathsf{pf}(\pi^{\star}) \models \mathrm{B}$, so, there exists a finite set $\mathrm{R} \subseteq \mathrm{Rq}(c_{\pi^{\star}})$ such that $\mathrm{Rq}(c_{\pi^{\star}}) \setminus \mathrm{R} \subseteq \mathrm{Rs}(c_{\pi^{\star}})$ with $c_{\pi^{\star}} = \mathsf{pf}(\pi^{\star})$. Now, construct a strategy $\sigma_{\exists} \in \mathrm{Str}_{\exists}(\Game)$ such that, for all $\pi \in \mathrm{Pth}(\Game_{\restriction \sigma_{\exists}})$, there exists $\pi^{\star} \in \mathrm{Pth}(\Game^{\star}_{\restriction \sigma_{\exists^{\star}}})$, with $\pi = \pi^{\star}_{\restriction 1}$. To do this, let $ext : \mathrm{Hst}_{\exists} \longrightarrow \mathrm{R}_{\perp}$ be a function mapping each history $\rho \in \mathrm{Hst}_{\exists}(\Game)$ to the biggest color request not yet answered along a play or to $\perp$, in case there are not unanswered requests. So, we set $\sigma_{\exists}(\rho) \triangleq \sigma^{\star}_{\exists}((\mathsf{lst}(\rho), ext(\rho)))_{\restriction 1}$, for all $\rho \in \mathrm{Hst}_{\exists}(\Game)$. At this point, for each strategy $\sigma_{\forall} \in \mathrm{Str}_{\forall}(\Game)$, there is a strategy $\sigma^{\star}_{\forall} \in \mathrm{Str}_{\forall}(\Game^{\star})$ such that $c_{\pi} \triangleq \mathsf{pf}(v_{0}, (\sigma_{\exists}, \sigma_{\forall})) \in$ FP, $c_{\pi^{\star}} \triangleq \mathsf{pf}(v_{0}, (\sigma^{\star}_{\exists}, \sigma^{\star}_{\forall})) \in \mathrm{B}$ and $c_{\pi} = (c_{\pi^{\star}})_{\restriction 1}$. Set $\sigma^{\star}_{\forall}$ using $\sigma_{\forall}$ as follows: $\sigma^{\star}_{\forall}((v, r)) = \sigma_{\forall}((v, r'))$ where $r' = \mathsf{tr}(r, \mathsf{cl}(v))$. Since $\mathsf{pf}(\pi^{\star}) \models \mathrm{B}$, we have that $c_{\pi^{\star}} \in (\mathrm{Cl}^{*} \cdot 2)^{\omega}$. Due to the structure of the transition table and the fact that we give a priority 2 to the vertexes in which there are not unanswered requests, we have that $\mathrm{Rq}(c_{\pi^{\star}}) = \mathrm{Rs}(c_{\pi^{\star}})$ and so $\mathrm{Rq}(c_{\pi}) = \mathrm{Rs}(c_{\pi})$.

[**Only If**] By hypothesis, we have that player $\exists$ wins the game $\Game$ on the weighted arena $\overline{\mathcal{A}}$ which induces a payoff arena $\widehat{\mathcal{A}}$. This means that, there exists a strategy $\sigma_{\exists} \in \mathrm{Str}_{\exists}(\Game)$ for player $\exists$ such that for each strategy $\sigma_{\forall} \in \mathrm{Str}_{\forall}(\Game)$ for player $\forall$, it holds that $\mathsf{pf}(v_{0}, (\sigma_{\exists}, \sigma_{\forall})) \in$ FP. Therefore, for all $\pi \in \mathrm{Pth}(\Game_{\restriction \sigma_{\exists}})$, we have that $\mathsf{pf}(\pi) \models$ FP, so, $\mathrm{Rq}(c_{\pi}) = \mathrm{Rs}(c_{\pi})$ with $c_{\pi} = \mathsf{pf}(\pi)$. Now, we construct a strategy $\sigma^{\star}_{\exists} \in \mathrm{Str}_{\exists}(\Game^{\star})$ for player $\exists$ on $\overline{\mathcal{A}^{\star}}$ as follows: for all vertexes (v, r), where $r \in \mathrm{R}_{\perp}$, it holds that $\sigma^{\star}_{\exists}(v, r) = \sigma_{\exists}(v)$. We prove that $\mathsf{pf}(\pi^{\star}) \models \mathrm{B}$ for all play $\pi^{\star} \in \mathrm{Pth}(\Game^{\star}_{\restriction \sigma^{\star}_{\exists}})$, *i.e.*, there exists a finite set $\mathrm{R} \subseteq \mathrm{Rq}(c_{\pi^{\star}})$ such that $\mathrm{Rq}(c_{\pi^{\star}}) \setminus \mathrm{R} \subseteq \mathrm{Rs}(c_{\pi^{\star}})$ with $c_{\pi^{\star}} = \mathsf{pf}(\pi^{\star})$. To do this, we project out π from $\pi^{\star}$, *i.e.*, $\pi = \pi^{\star}_{\restriction 1}$, whose meaning is $\pi^{\star}_{i} = (\pi_{i}, r_{i})$, for all $i \in \mathbb{N}$. . It easy to see that $\pi \in \mathrm{Pth}(\Game_{\restriction \sigma_{\exists}})$ and then $\mathsf{pf}(\pi) \models$ FP. By contradiction, assume that $\mathsf{pf}(\pi^{\star}) \not\models \mathrm{B}$. Consequently, there are no vertexes $(v, \perp)$ that appear infinitely often. This means that there exists a position $i \in \mathbb{N}$ in which there is a request $r \in \mathrm{Rq}(c_{\pi})$ not satisfied. But this means $\mathsf{pf}(\pi) \not\models$ FP, which is impossible. ☐

In the following, we report some examples of arenas obtained applying the reduction mentioned above. Observe that each vertex of the constructed arena is labeled with its name (in the upper part) and, in according to the transition function, by the biggest request not responded (in the middle part) and its color (in the lower part).

Example 5.2. Consider, now, the colored arena depicted in Figure 10. It represents the reduction from the colored arena drawn in Figure 2 where player $\exists$ wins the FP game $\Game$ because all requests are responded. Moreover, as previously proved, player $\exists$ wins also the B game $\Game^{\star}$ in Figure 10, obtained by the same colored arena $\widetilde{\mathcal{A}}$, visiting infinitely often the vertex $\mathtt{v}_{0}$ having priority 2.

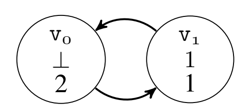

Figure 10: From Full Parity to Buchi.

Example 5.3. Consider, now, the colored arena depicted in Figure 11. It represents a reduction from the colored arena drawn in Figure 5. Here, we have that player $\exists$ loses the FP game $\Game$ because we have that the request at the vertex $\mathtt{v}_{0}$ is not responded. Also, as previously proved, he loses the B game $\Game^{\star}$ on the colored arena $\widetilde{\mathcal{A}}$, visiting infinitely often the vertex $\mathtt{v}_{1}$ having priority 1.

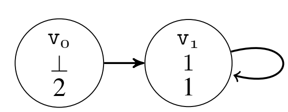

Figure 11: From Full Parity to Buchi.

5.3. From Bounded-Cost Parity to Parity

In this section, we show a construction that allows to reduce a bounded-cost parity game to a parity game. The approach we propose extends the one given in the previous section by further equipping the transition table $\mathcal{T}$ with a counter that keeps track of the delay accumulated since an unanswered request has been issued. Such a counter is bounded in the sense that if the delay exceeds the sum of weights of all moves in the original arena, then it is set to the special symbol $\circledast$. The idea is that if in a finite game such a bound has been exceeded then the adversarial player has taken at least twice a move with a positive weight. So, he can do this an arbitrary number of times and delay longer and longer the satisfaction of a request that therefore becomes not prompt. Thus, we use as states in $\mathcal{T}$, together with $\circledast$, a finite set of pairs of numbers, where the first component, as above, represents a finite request, while the second one is its delay. As first state component we also allow $\bot$, since with $(\bot, 0)$ we indicate the fact that there are not unanswered requests up to the current position. Then, the transition function of $\mathcal{T}$ is defined as follows. If a request is not satisfied within a bounded delay, then it goes and remains forever in state $\circledast$. Otherwise, if the request is satisfied, then it goes to $(\bot, 0)$, else it moves to a state that contains, as first component, the maximum between the last request not responded and the read color and, as second component, the one present in the current state plus the weight of the traversed edge.

Now, consider the product arena $\mathcal{A}^\star$ of $\mathcal{T}$ with the original arena and color its positions as follows: unanswered request positions, with delay exceeding the bound, are colored with 1, while the remaining ones are colored as in the original arena. Clearly, in $\mathcal{A}^\star$, a parity game is won if and only if the bounded-cost parity game is won on the original arena. The formal construction of $\mathcal{T}$ and $\mathcal{A}^\star$ follow.

For a given BCP game $\Game \triangleq \langle \widehat{\mathcal{A}}, \mathrm{BCP}, v_{\mathrm{o}} \rangle$ induced by a weighted arena $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$, we construct a deterministic transition table $\mathcal{T} \triangleq \langle \mathrm{Cl} \times \mathrm{Wg}, \mathrm{St}, \mathsf{tr} \rangle$, with set of states $\mathrm{St} \triangleq \{\circledast\} \cup \mathrm{R}_\bot \times [0, s]$, where we assume $s \triangleq \sum_{m \in Mv} \mathsf{wg}(m)$ to be the sum of all weights of moves in $\overline{\mathcal{A}}$, and transition function defined as follows: $\mathsf{tr}(\circledast, (c, w)) \triangleq \circledast$ and, additionally,

- $$\mathsf{tr}((r, k), (c, w)) \triangleq \begin{cases} (\bot, 0), & \text{if } r < c \text{ and } c \equiv 0 (\mathrm{mod}\ 2); \\ \circledast, & \text{if } k + w > s; \\ (\max\{r, c\}, k + w), & \text{otherwise.} \end{cases}$$

Let $\mathcal{A}^\star = \widetilde{\mathcal{A}} \otimes \mathcal{T}$ be the product arena of $\widetilde{\mathcal{A}}$ and $\mathcal{T}$, and $\widetilde{\mathcal{A}^\star} \triangleq \langle \mathcal{A}^\star, \mathrm{Cl}, \mathsf{cl}^\star \rangle$ the colored arena such that the state $(\mathrm{v}, \circledast)$ is colored with 1, while all other states are colored as in the original arena (w.r.t. the first component). Then, the P game $\Game^\star = \langle \widehat{\mathcal{A}^\star}, \mathrm{P}, (v_{\mathrm{o}}, (\bot, 0)) \rangle$ induced by $\widetilde{\mathcal{A}^\star}$ is such that player $\exists$ wins $\Game$ iff it wins $\Game^\star$.

Theorem 5.4. For every finite BCP game $\Game$ with $k \in \mathbb{N}$ priorities and sum of weights $s \in \mathbb{N}$, there is a P game $\Game^\star$, with order $|\Game^\star| = \mathrm{O}(|\Game| \cdot k \cdot s)$, such that player $\exists$ wins $\Game$ iff it wins $\Game^\star$.

Proof:
[If] By hypothesis, player $\exists$ wins the game $\Game^\star$ on the colored arena $\widetilde{\mathcal{A}}$, which induces a payoff arena $\widehat{\mathcal{A}}$. This means that there exists a strategy $\sigma_\exists^\star \in \mathrm{Str}_\exists(\Game^\star)$ for player $\exists$ such that for each strategy $\sigma_\forall^\star \in \mathrm{Str}_\forall(\Game^\star)$ for player $\forall$, it holds that $\mathsf{pf}(v_{\mathrm{o}}, (\sigma_\exists^\star, \sigma_\forall^\star)) \in \mathrm{P}$. Therefore, for all $\pi^\star \in \mathrm{Pth}(\Game^\star_{\restriction \sigma_\exists^\star})$, we have that $\mathsf{pf}(\pi^\star) \models \mathrm{P}$, hence, there exists a finite set $\mathrm{R} \subseteq \mathrm{Rq}(c_{\pi^\star})$ such that $\mathrm{Rq}(c_{\pi^\star}) \setminus \mathrm{R} \subseteq \mathrm{Rs}(c_{\pi^\star})$ with

$c_{\pi^\star} = \mathsf{pf}(\pi^\star)$. Now, we construct a strategy $\sigma_\exists \in \mathrm{Str}_\exists(\Game)$ such that, for all $\pi \in \mathrm{Pth}(\Game_{\restriction\sigma_\exists})$, there exists $\pi^\star \in \mathrm{Pth}(\Game^\star_{\restriction\sigma_\exists{}^\star})$, *i.e.*, $\pi = \pi^\star_{\restriction 1}$. Let $ext : \mathrm{Hst}_\exists \longrightarrow (\mathrm{R}_\bot \times \mathbb{N})$ be a function mapping each history $\rho \in \mathrm{Hst}_\exists(\Game)$ to a pair of values representing, respectively, the biggest (color) request not yet answered along the history and the sum of the weights over the crossed edges, from the last response of the request. So, we set $\sigma_\exists(\rho) \triangleq \sigma^\star_\exists((\mathsf{lst}(\rho), ext(\rho)))_{\restriction 1}$, for all $\rho \in \mathrm{Hst}_\exists(\Game)$. At this point, for each strategy $\sigma_\forall \in \mathrm{Str}_\forall(\Game)$, there is a strategy $\sigma^\star_\forall \in \mathrm{Str}_\forall$ such that $(c_\pi, w_\pi) \triangleq \mathsf{pf}(v_0, (\sigma_\exists, \sigma_\forall)) \in \mathrm{BCP}$, $c_{\pi^\star} \triangleq \mathsf{pf}(v_0, (\sigma^\star_\exists, \sigma^\star_\forall)) \in \mathrm{P}$ and $c_\pi = (c_{\pi^\star})_{\restriction 1}$. Set $\sigma^\star_\forall$ using, trivially, $\sigma_\forall$ as follows: $\sigma^\star_\forall((\mathtt{v}, (r, k))) = (\sigma_\forall(\mathtt{v}), (r^{'}, k^{'}))$ where $(r', k') = \mathsf{tr}((r, k), (\mathsf{cl}(\mathtt{v}), \mathsf{wg}((\mathtt{v}, \sigma_\forall(\mathtt{v})))))$. Let $b = \max\{k \in \mathbb{N} \mid \exists\, i \in \mathbb{N}, \exists\, v \in \mathrm{St}(\Game), r \in \mathrm{R}_\bot.(\pi^\star)_i = (v, (r, k))\}$ be the maximum value the counter can have and $s = \sum_{e \in Mv} \mathsf{wg}(e)$ the sum of weights of edges over the weighted arena $\overline{\mathcal{A}}$. Since $\mathsf{pf}(\pi^\star) \models \mathrm{P}$, by construction, we have that there is no state $(\mathtt{v}, \divideontimes)$ in $\pi^\star$. Moreover, all states $(\mathtt{v}, (r, k))$ in $\pi^\star$ have $k \leq b \leq s$. In other words, b corresponds to the delay within which the request is satisfied. Thus, there exists both a finite set $\mathrm{R} \subseteq \mathrm{Rq}(c_\pi)$ such that $\mathrm{Rq}(c_\pi) \setminus \mathrm{R} \subseteq \mathrm{Rs}(c_\pi)$ and a bound $b \in \mathbb{N}$ for which $\mathrm{dl}((c_\pi, w_\pi), \mathrm{Rq}(c_\pi)) \leq b$.

[**Only If**] By hypothesis, player $\exists$ wins the game $\Game$ on the weighted arena $\overline{\mathcal{A}}$, which induces a payoff arena $\widehat{\mathcal{A}}$. This means that there exists a strategy $\sigma_\exists \in \mathrm{Str}_\exists(\Game)$ for player $\exists$ such that for each strategy $\sigma_\forall \in \mathrm{Str}_\forall(\Game)$ for player $\forall$, it holds that $\mathsf{pf}(v_0, (\sigma_\exists, \sigma_\forall)) \in \mathrm{BCP}$. Therefore, for all $\pi \in \mathrm{Pth}(\Game_{\restriction\sigma_\exists})$, we have that $\mathsf{pf}(\pi) \models \mathrm{BCP}$. Hence, there exists a finite set $\mathrm{R} \subseteq \mathrm{Rq}(c_\pi)$ such that $\mathrm{Rq}(c_\pi) \setminus \mathrm{R} \subseteq \mathrm{Rs}(c_\pi)$ and a bound $b \in \mathbb{N}$ for which $\mathrm{dl}((c_\pi, w_\pi), \mathrm{Rq}(c_\pi)) \leq b$, where $(c_\pi, w_\pi) = \mathsf{pf}(\pi)$. Let s be the sum of weights of edges in the original arena $\overline{\mathcal{A}}$, previously defined. Now, we construct a strategy $\sigma^\star_\exists \in \mathrm{Str}_\exists(\Game^\star)$ for player $\exists$ on $\overline{\mathcal{A}^\star}$ as follows: for all vertexes $(v, (r, k))$, where $r \in \mathrm{R}_\bot$ and $k \in [0, s]$, it holds that $\sigma^\star_\exists(((v, (r, k)))) = (\sigma_\exists(v), (r^{'}, k^{'}))$ where $(r^{'}, k^{'}) = \mathsf{tr}((r, k), (\mathsf{cl}(\mathtt{v}), \mathsf{wg}((\mathtt{v}, \sigma_\exists(\mathtt{v})))))$. We want to prove that $\mathsf{pf}(\pi^\star) \models \mathrm{P}$, for all plays $\pi^\star \in \mathrm{Pth}(\Game^\star_{\restriction\sigma^\star_\exists})$, *i.e.*, there exists a finite set $\mathrm{R} \subseteq \mathrm{Rq}(c_{\pi^\star})$ such that $\mathrm{Rq}(c_{\pi^\star}) \setminus \mathrm{R} \subseteq \mathrm{Rs}(c_{\pi^\star})$ with $c_{\pi^\star} = \mathsf{pf}(\pi^\star)$. To do this, first suppose that, for all plays $\pi^\star \in \mathrm{Pth}(\Game^\star_{\restriction\sigma^\star_\exists})$, $\pi^\star$ does not cross a state of the kind $(\mathtt{v}, \divideontimes) \in \mathrm{St}(\Game^\star)$ and projects out π from $\pi^\star$, *i.e.*, $\pi = \pi^\star_{\restriction 1}$. It easy to see that $\pi \in \mathrm{Pth}(\Game_{\restriction\sigma_\exists})$ and, so, $\mathsf{pf}(\pi) \models \mathrm{BCP}$. Consequently, $\mathsf{pf}(\pi) \models \mathrm{P}$. Now, due to our assumption, the colors in $\mathsf{pf}(\pi)$ and $\mathsf{pf}(\pi^\star)$ are the same, *i.e.*, $c_\pi = c_{\pi^\star}$. Thus, it holds that $\mathsf{pf}(\pi^\star) \models \mathrm{P}$. It remains to see that our assumption is the only possible one, *i.e.*, it is impossible to find a path $\pi^\star \in \mathrm{Pth}(\Game^\star_{\restriction\sigma^\star_\exists})$, containing a state of the the kind $(\mathtt{v}, \divideontimes) \in \mathrm{St}(\Game^\star)$. By contradiction, assume that there exists a position $i \in \mathbb{N}$ in which there is a request $r \in \mathrm{Rq}(c_{\pi^\star}) \setminus \mathrm{R}$ not satisfied within delay at most s. Moreover, let j be the first position in which a state of kind $(\mathtt{v}, \divideontimes)$ is traversed. Between the states $(\mathtt{v}_i, (r_i, k_i)) = (\pi^\star)_i$ and $(\mathtt{v}_j, (r_j, k_j)) = (\pi^\star)_j$, there are no states whose color is an even number bigger than $\mathsf{cl}(\mathtt{v}_i)$. Then, it holds that $\sum_{h=i}^{j} \mathsf{wg}(h) > s$, *i.e.*, at least one of the edges is repeated. Let l and $l^{'}$ with $l < l^{'}$ be two positions in π in which the same edge is repeated, *i.e.*, $(\pi_l, \pi_{l+1}) = (\pi_{l^{'}}, \pi_{l^{'}+1})$. Observe that $\mathsf{wg}((\pi_{l^{'}}, \pi_{l^{'}+1})) > 0$ since otherwise we would not have exceeded the bound s. Furthermore, $\pi_{l+1} = \pi_{l+1^{'}}$ is necessarily a state of player $\forall$. So, he has surely a strategy forcing the play π to pass infinitely often through the edge $(\pi_{l^{'}}, \pi_{l^{'}+1})$. This means that $\mathsf{pf}(\pi) \not\models \mathrm{BCP}$, which is impossible.

$\square$

In the following, we report some examples of arenas obtained applying the reduction mentioned above. Observe that, each vertex of the constructed arena is labeled with its name (in the upper part) and, in according to the transition function, by the pair containing the biggest request not responded and the counter from the last request not responded (in the middle part) and its color (in the lower part).

Example 5.5. Consider the weighted arena depicted in Figure 12. It represents the reduction from the arena drawn in Figure 6. In this example, player $\exists$ wins the BCP $\Game$ because the request at the vertex $\mathtt{v}_0$ is not responded but it has a bounded delay equals to 1. Moreover, as previously showed, player $\exists$ wins also the P game $\Game^\star$ obtained from the same weighted arena $\overline{\mathcal{A}}$, visiting infinitely often the vertex $(\mathtt{v}_1, (1, 1))$ having priority 0.

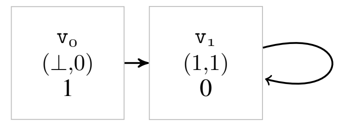

Figure 12: From Bounded-Cost Parity to Parity.

Example 5.6. Consider the weighted arena in Figure 13. It represents the reduction from the arena drawn in Figure 3. In this example, player $\exists$ loses the BCP $\Game$ because the request at the vertex $\mathtt{v}_0$ is not responded and there is a unique play in which the delay is incremented by 1 in unbounded way. Moreover, as previously showed, player $\exists$ loses also the P game $\Game^\star$ obtained from the same weighted arena $\overline{\mathcal{A}}$, in which there exists a unique play where the special states $(\mathtt{v}_2, \ast)$ and $(\mathtt{v}_1, \ast)$ with priority 1, are visited infinitely often.

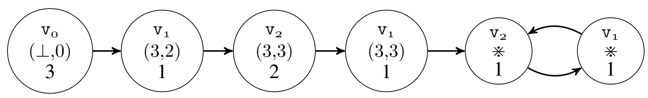

Figure 13: From Bounded-Cost Parity to Parity.

5.4. From Prompt Parity to Parity and Büchi

Finally, we show a construction that reduces a prompt parity game to a parity game. In particular, when the underlying weighted arena of the original game has only positive weights, then the construction returns a Büchi game. Our approach extends the one proposed for the above BCP case, by further allowing the transition table $\mathcal{T}$ to guess a request value that is not meet anymore along a play. This is done to accomplish the second part of the prompt parity condition, in which a finite number of requests can be excluded from the delay computation. To do this, first we allow $\mathcal{T}$ to be nondeterministic and label its states with a flag $\alpha \in \{D, \exists\}$ to identify, respectively, deterministic and existential states. Then, we enrich the states by means of a new component $d \in [0, h]$, where $h \triangleq |\{v \in \mathrm{Ps} : \mathsf{cl}(v) \equiv 1 \pmod 2\}|$ is the maximum number of positions having odd priorities. So, d represents the counter of the forgotten priority, which it is used to later check the guess states. The existential states belong to player $\exists$. Conversely, the deterministic ones belong to player $\forall$. As initial state we have the tuple $(\mathtt{v}, ((\bot, 0, D), 0)))$ indicating that there are not unanswered and forgotten requests up to the current deterministic position. The transition function over a deterministic state is defined as follows. If a request is not satisfied in a bounded delay, (*i.e.*, the delay exceeds the sum of the weights of all moves in the original arena) then it goes and remains forever in state $(\mathtt{v}, \ast)$ with priority 1; if the request is satisfied then it goes to $(\mathtt{v}, ((\bot, d, D), 0))$ indicating that in this deterministic state there is not an unanswered request and the sum of the weight of the edges is 0); otherwise it moves to an existential state that contains, as first component, the triple having the maximum between the last request not responded and the read color, the counter of forgotten priority, and a flag indicating that the state is existential. Moreover, as a second component, there is a number that represents the sum of the weights of the traversed edges until the current state. The transition function over an existential state is defined as follows. If d is equal to h (*i.e.*, the maximum allowable number of positions having an odd priority), then the computation remains in the same (deterministic) state;

otherwise, the computation moves to a state in which the second component is incremented by the weight of the crossed edge. Note that the guess part is similar to that one performed to translate a nonderministic co-Büchi automaton into a Büchi one [21]. Finally, we color the positions of the obtained arena as follows: unanswered request positions, with delay exceeding the bound, are colored by 1, while the remaining ones are colored as in the original arena. In case the weighted arena of the original game has only positive weights, then one can exclude a priory the fact that there are unanswered requests with bounded delays. So, all these kind of requests can be forgotten in order to win the game. Thus, in this case, it is enough to satisfy only the remaining ones, which corresponds to visit infinitely often a position containing as second component the symbol $\bot$. So it is enough to color these positions with 2, all the remaining ones with 1, and play on this arena a Büchi condition. The formal construction of the transition table and the enriched arena follow.

For a PP game $\Game \triangleq \langle \widehat{\mathcal{A}}, \mathrm{PP}, v_{0} \rangle$ induced by an arena $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$, we build a transition table $\mathcal{T} \triangleq \langle \mathrm{Cl} \times \mathrm{Wg}, \mathrm{St}_{D}, \mathrm{St}_{\exists}, \mathsf{tr} \rangle$, with sets of states $\mathrm{St}_{D} \triangleq \{\divideontimes\} \cup \mathrm{Z}_{D} \times [0, s]$ and $\mathrm{St}_{\exists} \triangleq \mathrm{Z}_{\exists} \times [0, s]$, where we assume $s \triangleq \sum_{m \in Mv} \mathsf{wg}(m)$ to be the sum of all weights of moves in the original arena and $\mathrm{Z}_{\alpha} \triangleq \mathrm{R}_{\bot} \times [0, h] \times \alpha$, and its transition function defined as follows: $\mathsf{tr}(\divideontimes, (c, w)) \triangleq \divideontimes$ and, additionally:

- $\mathsf{tr}(((r, d, D), k), (c, w)) \triangleq \begin{cases} ((\bot, d, D), 0), & \text{if } r < c \wedge c \equiv 0 \,(\mathrm{mod}\ 2); \\ \divideontimes, & \text{if } k + w > s; \\ ((\max\{r, c\}, d, \exists), k + w), & \text{otherwise.} \end{cases}$

- $\mathsf{tr}(((r, d, \exists), k)) \triangleq \begin{cases} \{((r, d, D), k)\}, & \text{if } d = h; \\ \{((r, d, D), k), ((\bot, d + 1, D), 0)\}, & \text{otherwise.} \end{cases}$

Observe that, the set Z_{α} is the Cartesian product of the biggest unanswered request, the counter of the forgotten priority and, a flag indicating whether the state is deterministic or existential.

Let $\mathcal{A}^{\star} = \overline{\mathcal{A}} \otimes \mathcal{T}$ be the product arena of $\overline{\mathcal{A}}$ and $\mathcal{T}$ and consider the colored arena $\widetilde{\mathcal{A}^{\star}} \triangleq \langle \mathcal{A}^{\star}, \mathrm{Cl}, \mathsf{cl}^{\star} \rangle$ such that, for all positions $(v, t) \in \mathrm{Ps}^{\star}$, if $t = \divideontimes$ then $\mathsf{cl}^{\star}((v, t)) = 1$ else $\mathsf{cl}^{\star}((v, t)) = \mathsf{cl}(v)$. Then, the P game $\Game^{\star} = \langle \widehat{\mathcal{A}^{\star}}, \mathrm{P}, (v_{0}, ((\bot, 0, D), 0)) \rangle$ induced by $\widetilde{\mathcal{A}^{\star}}$ is such that player $\exists$ wins $\Game$ iff it wins $\Game^{\star}$.

Theorem 5.7. For every PP game $\Game$ with $k \in \mathbb{N}$ priorities and sum of weights $s \in \mathbb{N}$, there is a P game $\Game^{\star}$, with order $|\Game^{\star}| = \mathrm{O}(|\Game|^{2} \cdot k \cdot s)$, such that player $\exists$ wins $\Game$ iff it wins $\Game^{\star}$.

Proof:
[If] By hypothesis, player $\exists$ wins the game $\Game^{\star}$ on the colored arena $\widetilde{\mathcal{A}}$, which induces a payoff arena $\widehat{\mathcal{A}}$. This means that there exists a strategy $\sigma_{\exists}^{\star} \in \mathrm{Str}_{\exists}(\Game^{\star})$ for player $\exists$ such that for each strategy $\sigma_{\forall}^{\star} \in \mathrm{Str}_{\forall}(\Game^{\star})$ for player $\forall$, it holds that $\mathsf{pf}(v_{0}, (\sigma_{\exists}^{\star}, \sigma_{\forall}^{\star})) \in \mathrm{P}$. Therefore, for all $\pi^{\star} \in \mathrm{Pth}(\Game^{\star}_{\restriction \sigma_{\exists}^{\star}})$, we have that $\mathsf{pf}(\pi^{\star}) \models \mathrm{P}$. Hence, there exists a finite set $\mathrm{R} \subseteq \mathrm{Rq}(c_{\pi^{\star}})$ such that $\mathrm{Rq}(c_{\pi^{\star}}) \setminus \mathrm{R} \subseteq \mathrm{Rs}(c_{\pi^{\star}})$ with $c_{\pi^{\star}} = \mathsf{pf}(\pi^{\star})$. Now, we construct a strategy $\sigma_{\exists} \in \mathrm{Str}_{\exists}(\Game)$ such that, for all $\pi \in \mathrm{Pth}(\Game_{\restriction \sigma_{\exists}})$, there exists $\pi^{\star} \in \mathrm{Pth}(\Game^{\star}_{\restriction \sigma_{\exists}^{\star}})$, *i.e.*, $\pi = \pi^{\star}_{\restriction 1}$. Let $ext : \mathrm{Hst}_{\exists} \longrightarrow (\mathrm{R}_{\bot} \times [0, h] \times D) \times \mathbb{N}$ be a function mapping each history $\rho \in \mathrm{Hst}_{\exists}(\Game)$ to a tuple of values that represent, respectively, the biggest color request along the history ρ that is both not answered and not forget by ${\sigma_{\exists}}^{\star}$, the number of odd priorities that are forgotten, and the sum of the weights over the crossed edges since the more recent occurrence of one of the following two cases: the last response of a request or the last request that is forgotten. So, we set $\sigma_{\exists}(\rho) \triangleq \sigma_{\exists}^{\star}((\mathsf{lst}(\rho), ext(\rho)))_{\restriction 1}$, for all $\rho \in \mathrm{Hst}_{\exists}(\Game)$. At this point, for each

strategy $\sigma_{\forall} \in \mathrm{Str}_{\forall}(\Game)$, there is a strategy $\sigma^{\star}_{\forall} \in \mathrm{Str}_{\forall}$ such that $(c_{\pi}, w_{\pi}) \triangleq \mathsf{pf}(v_0, (\sigma_{\exists}, \sigma_{\forall})) \in \mathrm{PP}$, $c_{\pi^{\star}} \triangleq \mathsf{pf}(v_0, (\sigma^{\star}_{\exists}, \sigma^{\star}_{\forall})) \in \mathrm{P}$ and $c_{\pi} \triangleq (c_{\pi^{\star\prime}})_{\restriction 1}$ where $\pi^{\star'}$ is obtained from $\pi^{\star}$ by removing the vertexes of the form $(v, ((r, d, \exists), k))$ that are the vertexes in which it is allowed to forget a request. Now, set $\sigma^{\star}_{\forall}$ using $\sigma_{\forall}$ as follows: $\sigma^{\star}_{\forall}(\mathtt{v}, ((r, d, \alpha), k)) = (\sigma_{\forall}(\mathtt{v}), ((r', d', \alpha'), k'))$ where $((r', d', \alpha'), k') = \mathsf{tr}(((r, d, \alpha), k), (\mathsf{cl}(\mathtt{v}), \mathsf{wg}((\mathtt{v}, \sigma_{\forall}(\mathtt{v})))))$. Let $b = \max\{k \in \mathbb{N} \mid \exists\, i \in \mathbb{N}, \exists\, v \in \mathrm{St}(\Game), r \in \mathrm{R}_{\perp}, d \in [0, h], \alpha \in \{D, \exists\}.(\pi^{\star})_i = (v, ((r, d, \alpha), k))\}$ be the maximum value that the counter can have and $s = \sum_{e \in Mv} \mathsf{wg}(e)$ the sum of weights of edges over the weighted arena $\overline{\mathcal{A}}$. Since $\mathsf{pf}(\pi^{\star}) \models \mathrm{P}$, by construction we have that there is no state $(\mathtt{v}, \circledast)$ in $\pi^{\star}$. Moreover, all states $(\mathtt{v}, ((r, d, \alpha), k))$ in $\pi^{\star}$ have $k \leq b \leq s$. Thus, there exists both a finite set $\mathrm{R} \subseteq \mathrm{Rq}(c_{\pi})$ such that $\mathrm{Rq}(c_{\pi}) \setminus \mathrm{R} \subseteq \mathrm{Rs}(c_{\pi})$ and a bound $b \in \mathbb{N}$ for which $\mathrm{dl}((c_{\pi}, w_{\pi}), \mathrm{Rq}(c_{\pi}) \setminus \mathrm{R}) \leq b$.

[**Only If**] By hypothesis, player $\exists$ wins the game $\Game$ on the weighted arena $\overline{\mathcal{A}}$, which induces a payoff arena $\widehat{\mathcal{A}}$. This means that there exists a strategy $\sigma_{\exists} \in \mathrm{Str}_{\exists}(\Game)$ for player $\exists$ such that, for each strategy $\sigma_{\forall} \in \mathrm{Str}_{\forall}(\Game)$ for player $\forall$, it holds that $\mathsf{pf}(v_0, (\sigma_{\exists}, \sigma_{\forall})) \in \mathrm{PP}$. Therefore, for all $\pi \in \mathrm{Pth}(\Game_{\restriction \sigma_{\exists}})$ we have that $\mathsf{pf}(\pi) \models \mathrm{PP}$. Hence, there exists a finite set $\mathrm{R} \subseteq \mathrm{Rq}(c_{\pi})$ such that $\mathrm{Rq}(c_{\pi}) \setminus \mathrm{R} \subseteq \mathrm{Rs}(c_{\pi})$ and there exists a bound $b \in \mathbb{N}$ for which $\mathrm{dl}((c_{\pi}, w_{\pi}), \mathrm{Rq}(c_{\pi}) \setminus \mathrm{R}) \leq b$, with $(c_{\pi}, w_{\pi}) = \mathsf{pf}(\pi)$. Let $h \triangleq |\{v \in \mathrm{Ps} : \mathsf{cl}(v) \equiv 1 \,(\mathrm{mod}\ 2)\}|$ be the maximum number of positions having odd priorities. Moreover, let s be the sum of all weights of moves in the original game $\Game$, previously defined. Now, we construct a strategy $\sigma^{\star}_{\exists} \in \mathrm{Str}_{\exists}(\Game^{\star})$ for player $\exists$ on $\overline{\mathcal{A}^{\star}}$ as follows. For all vertexes $(v, ((r, d, D), k)) \in \mathrm{St}_{\exists}(\Game^{\star})$, we set $\sigma^{\star}_{\exists}(v, ((r, d, D), k)) = (\sigma_{\exists}(v), ((r', d', \alpha'), k'))$ where $((r', d', \alpha'), k') = \mathsf{tr}(((r, d, D), k), (\mathsf{cl}(\mathtt{v}), \mathsf{wg}((\mathtt{v}, \sigma_{\exists}(\mathtt{v})))))$. Moreover, for all vertexes $(v, ((r, h, \exists), k))$ $\in \mathrm{St}_{\exists}(\Game^{\star})$, we set $\sigma^{\star}_{\exists}(v, ((r, h, \exists), k)) = (\sigma_{\exists}(v), ((r, h, D), k))$. Now, let $frg : \mathrm{St}_{\exists} \to \mathbb{N}$ be a function such that $frg(v)$ is the maximum odd priority that player $\exists$ can forget, *i.e.*, the highest odd priority that can be crossed only finitely often in $\Game_{\restriction \sigma_{\exists}}$ starting at v. At this point, if $d < h$, *i.e.*, it is still possible to forget other $h - d$ priorities, then we set $\sigma^{\star}_{\exists}(v, ((r, d, \exists), k)) = (\sigma_{\exists}(v), ((\perp, d + 1, D), 0))$ if $r \leq frg(v)$, otherwise, $\sigma^{\star}_{\exists}(v, ((r, d, \exists), k)) = (\sigma_{\exists}(v), ((r, d, D), k))$. We want to prove that $\mathsf{pf}(\pi^{\star}) \models \mathrm{P}$, for all play $\pi^{\star} \in \mathrm{Pth}(\Game^{\star}_{\restriction \sigma^{\star}_{\exists}})$, *i.e.*, there exists a finite set $\mathrm{R} \subseteq \mathrm{Rq}(c_{\pi^{\star}})$ such that $\mathrm{Rq}(c_{\pi^{\star}}) \setminus \mathrm{R} \subseteq \mathrm{Rs}(c_{\pi^{\star}})$ with $c_{\pi^{\star}} = \mathsf{pf}(\pi^{\star})$. Starting from $\pi^{\star}$, we construct $\pi^{\star'}$ by removing the vertexes of the form $(v, ((r, d, \exists), k))$ that are the vertexes in which is allow to forget a request . Then, we project out π from $\pi^{\star'}$, *i.e.*, $\pi = \pi^{\star'}_{\restriction 1}$. It easy to see that $\pi \in \mathrm{Pth}(\Game_{\restriction \sigma_{\exists}})$ and, so, $\mathsf{pf}(\pi) \models \mathrm{PP}$. Consequently, $\mathsf{pf}(\pi) \models \mathrm{P}$. The colors in $\mathsf{pf}(\pi)$ and $\mathsf{pf}(\pi^{\star'})$ are the same, *i.e.*, $c_{\pi} = c_{\pi^{\star\prime}}$. Thus, it holds that $\mathsf{pf}(\pi^{\star'}) \models \mathrm{P}$ and so $\mathsf{pf}(\pi^{\star}) \models \mathrm{P}$. At this point, it just remains to see that our assumption is the only possible one, *i.e.*, it is impossible to find a path $\pi^{\star} \in \mathrm{Pth}(\Game^{\star}_{\restriction \sigma^{\star}_{\exists}})$ containing a state of the the kind $(\mathtt{v}, \circledast) \in \mathrm{St}(\Game^{\star})$. To do this, we use the same reasoning applied in the proof of Theorem 5.4. □

It is worth observing that the estimation on the size of $\Game^{\star}$ in Theorem 5.7 is quite coarse since several type of states can not be reached by the initial position.

In the following, we report some examples of arenas obtained by applying the reduction mentioned above. Observe that each vertex of the constructed arena is labeled by its name (in the upper part) and, according to the transition function, by the tuple containing the biggest request not responded, the maximum number of forgotten positions having odd priorities in the original arena, a flag indicating a deterministic or an existential state, a counter from the last request not responded (in the middle part), and its color (in the lower part).

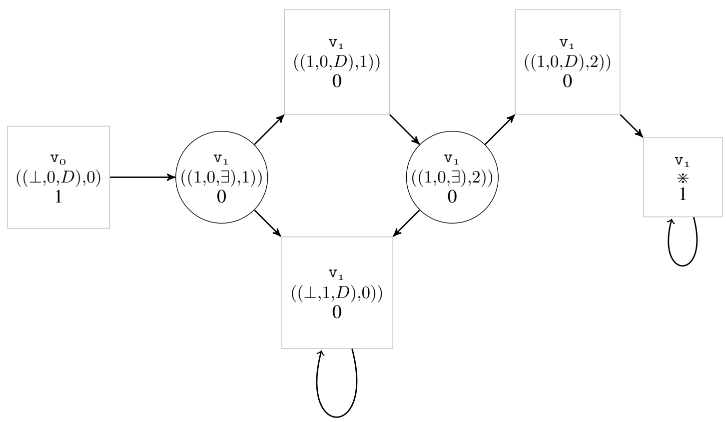

Figure 14: From Prompt Parity to Parity (Figura 5).

Example 5.8. Consider the weighted arena depicted in Figure 14. It represents the reduction from the arena drawn in Figure 5. In this example, player $\exists$ wins the PP game ⅁ because only the request at the vertex $\mathtt{v}_0$ is not responded and this request is not traversed infinitely often. Moreover, as previously showed, player $\exists$ also wins the P game ⅁* obtained from the same weighted arena in Figure 14. In more details, starting from the initial vertex $(\mathtt{v}_0, ((\bot, 0, D), 0))$ with priority 1, player $\forall$ moves the token to the existential vertex $(\mathtt{v}_1, ((1, 0, \exists), 1))$ having priority 0. At this point, player $\exists$ has two options: he can forget or not the biggest odd priority crossed up to now. In the first case, he moves to the vertex $(\mathtt{v}_1, ((\bot, 1, D), 0))$, having priority 0, where player $\forall$ can only cross infinitely often this vertex, letting player $\exists$ to win the game. In the other case, he moves to the vertex $(\mathtt{v}_1, ((1, 0, D), 1))$ with priority 0 from which player $\forall$ moves to the vertex $(\mathtt{v}_1, ((1, 0, \exists), 2))$ having priority 0. From this vertex, player $\exists$ can still decide either to forget or not the biggest odd priority crossed up to now. In the first case player $\exists$ wins the game by crossing infinitely often the vertex $(\mathtt{v}_1, ((\bot, 1, D), 0))$ with priority 0. In the other case, he loses the game and so he will never take such a move. In conclusion, player $\exists$ has a winning strategy against every possible strategy of the player $\forall$.

Example 5.9. Consider the weighted arena depicted in Figure 15. This arena represents the reduction from the arena in Figure 1. In this example, player $\exists$ loses the PP ⅁ against any possible move of the opponent because the delay between the request and its response is unbounded. Moreover, as proved, player $\exists$ loses also the P game ⅁* obtained from the same weighted arena $\overline{\mathcal{A}}$, against any possible move of the opponent. In detail, we have that the game starts in the vertex $(\mathtt{v}_0, ((\bot, 0, D), 0))$ having priority 1. At this point, player $\forall$ is obliged to go to the vertex $(\mathtt{v}_1, ((1, 0, \exists), 0))$ with priority 0. Then, player $\exists$ has two options that are either to forget or not forget the biggest odd priority crossed.

1. In the first case he goes to the vertex $(\mathtt{v}_1, ((\bot, 1, D), 0))$ having priority 0. From this vertex, player $\forall$, in order to avoid losing, does not cross this vertex infinitely often, but he moves the token in the vertex $(\mathtt{v}_2, ((\bot, 1, D), 0))$ having priority 2. From this vertex, player $\forall$ is obliged to move the token in the vertex $(\mathtt{v}_0, ((\bot, 1, D), 0))$ with priority 1 and yet to the vertex $(\mathtt{v}_1, ((1, 1, \exists), 0))$

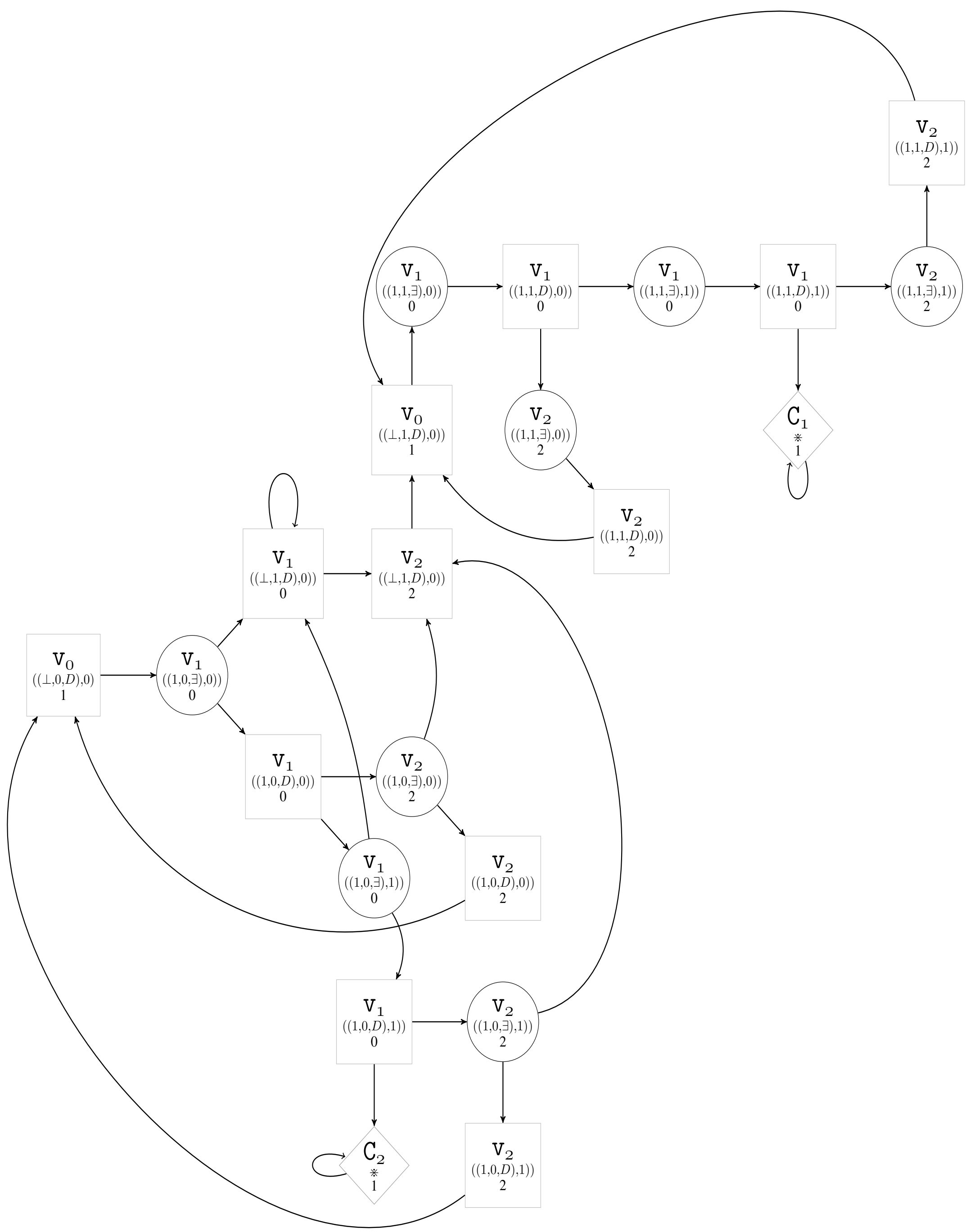

Figure 15: From Prompt Parity to Parity.

having priority 1. At this point, player $\exists$ can move the token only to the vertex $(\mathtt{v_1}, ((1, 1, D), 0))$ with priority 0, which belong to player $\forall$. Then, this player, moves the token in the vertex $(\mathtt{v_1}, ((1, 1, \exists), 1))$ with priority 0. From this vertex, player $\exists$ can only move the token to the vertex $(\mathtt{v_1}, ((1, 1, D), 1))$ from which player $\forall$ wins the game by forcing the token to remain in the diamond vertex $(\mathtt{C_1}, \circledast)$ which we use to succinctly represent a strong connected component, fully labeled by 1, from which player $\exists$ cannot exit.

2. In the other case, player $\exists$ goes to the vertex $(\mathtt{v_1}, ((1, 0, D), 0))$ having priority 0. At this point, player $\forall$ may decide to go either in the vertex $(\mathtt{v_2}, ((1, 0, \exists), 0))$ having priority 2 or in the vertex $(\mathtt{v_1}, ((1, 0, \exists), 1))$ with priority 0. From the vertex $(\mathtt{v_2}, ((1, 0, \exists), 0))$, player $\exists$ can decide either to forget or not the biggest odd priority crossed.

   (a) In the first case, player $\exists$ moves the token to the vertex $(\mathtt{v_2}, ((\bot, 1, D), 0))$ having priority 2 and the play continues as in step 1, starting from this vertex.

   (b) In the other case, player $\exists$ moves the token to the vertex $(\mathtt{v_2}, ((1, 0, D), 0))$ belonging to the player $\forall$ which moves the token at the initial vertex. At this point, player $\exists$ moves the token to the initial vertex $(\mathtt{v_0}, ((\bot, 0, D), 0))$ having priority 1. From this vertex, player $\forall$ goes to the vertex $(\mathtt{v_1}, ((1, 0, \exists), 0))$ with priority 0. Then, player $\exists$ can either forget or not the biggest odd priority crossed. From this state, we have already seen that he can win the game.

   From the vertex $(\mathtt{v_1}, ((1, 0, \exists), 1))$ with priority 0, player $\exists$ can:

   (a) decide to forget the biggest odd priority and then to move the token to the vertex $(\mathtt{v_1}, ((\bot, 1, D)$ $0))$ having priority 0. At this point, the play continues as in step 1 starting from this vertex.

   (b) decide to not forget the biggest odd priority and then to move the token to the vertex $(\mathtt{v_1}, ((1, 0, D), 1))$ belonging to the player $\forall$, which force the token to remain in the diamond vertex $(\mathtt{C_2}, \circledast)$ having priority 1, winning the game.

In case the weighted arena $\overline{\mathcal{A}}$ is positive, *i.e.*, $\mathsf{wg}(v) > 0$ for all $v \in \mathrm{Ps}$, we can improve the above construction as follows. Consider the colored arena $\widetilde{\mathcal{A}^\star} \triangleq \langle \mathcal{A}^\star, \{1, 2\}, \mathsf{cl}^\star \rangle$ such that, for all positions $(v, t) \in \mathrm{Ps}^\star$, if $t = ((\bot, d, D), 0)$ for some $d \in [0, h]$ then $\mathsf{cl}^\star((v, t)) = 2$ else $\mathsf{cl}^\star((v, t)) = 1$. Then, the B game $\Game^\star = \langle \widehat{\mathcal{A}^\star}, \mathrm{B}, (v_0, ((\bot, 0, D), 0)) \rangle$ induced by $\widetilde{\mathcal{A}^\star}$ is such that player $\exists$ wins $\Game$ iff it wins $\Game^\star$.

By means of a proof similar to the one used to prove Theorem 5.7 , we obtain the following.

Theorem 5.10. For every PP game $\Game$ with $k \in \mathbb{N}$ priorities and sum of weights $s \in \mathbb{N}$ defined on a positive weighted arena, there is a B game $\Game^\star$, with order $|\Game^\star| = \mathrm{O}(|\Game|^2 \cdot k \cdot s)$, such that player $\exists$ wins $\Game$ iff it wins $\Game^\star$.

6. Conclusion

Recently, promptness reasonings have received large attention in system design and verification. This is due to the fact that, while from a theoretical point of view questions like "a specific state is eventually reached in a computation" have a clear meaning and application in formal verification, in a practical scenario, such a question results useless if there is no bound over the time the required state occurs. This is the case, for example, when we deal with liveness and safety properties. The question becomes even more involved in the case of reactive systems, well modeled as two-player games, in which the response can be procrastinated later and later due to an adversarial behavior.

In this work, we studied several variants of two-player parity games working under a prompt semantics. In particular, we gave a general and clean setting to formally describe and unify most of such games introduced in the literature, as well as to address new ones. Our framework helped us to investigate peculiarities and relationships among the addressed games. In particular, it helped us to come up with solution algorithms that have as core engine and main complexity the solution of a parity or a Büchi game. This makes the proposed algorithms very efficient. With more details, we have considered games played over colored and weighted arenas. In colored arenas, vertexes are colored with priorities and the parity condition asks whether, along paths, every odd priority (a request) is eventually followed by a bigger even priority (a response). In addition, weighted arenas have weights over the edges and consider as a delay of a request the sum of the edges traversed until its response occurs. Also, we have differentiated conditions depending on whether it occurs not-full (all requests, but a finite number, have to be satisfied), full (all requests have to be satisfied) or semi-full (the condition is a conjunction of two request properties, one behaving full and the other not-full).

As games already addressed in the literature, we studied the cost parity and bounded-cost parity ones and, for both of them, we provided algorithms that improve their known complexity. As new parity games, we investigated the full parity, full-prompt parity, and prompt parity ones. We showed that full parity games are in PTIME, prompt parity and cost parity are equivalent and both in UPTIME ∩ COUPTIME. The latter improves the known complexity result of [17] to solve cost parity games because our algorithm reduces the original problem to a unique parity game, while the one in [17] performs "several calls" to a parity game solver. Tables 1 and 2 report the formal definition of all conditions addressed in the paper along with the full/not-full/semi-full behavior. Tables 3 summarizes the achieved results. In particular, we use the special arrow $\hookleftarrow$ to indicate that the result is trivial or an easy consequence of another one in the same row.

Conditions	Colored Arena	(Colored) Weighted arena
Parity (P)	UPTIME ∩ COUPTIME [19]	$\hookleftarrow$
Full Parity (FP)	PTIME [Thm 5.1]	$\hookleftarrow$
Prompt Parity (PP)	PTIME [Thm 5.10]	UPTIME ∩ COUPTIME [Thm 5.7]
Full Prompt Parity (FPP)	$\hookleftarrow$	PTIME [FP + Cor 4.2]
Cost Parity (CP)	PTIME [PP + Cor 4.3]	UPTIME ∩ COUPTIME [PP + Cor 4.3]
Bounded Cost Parity (BCP)	PTIME [FPP + Cor 4.4]	UPTIME ∩ COUPTIME [Thm 5.4]

Table 3: Summary of all winning condition complexities.

As future work, there are several directions one can investigate. For example, one can extend the same framework in the context of multi-agent systems. Recently, a (multi-agent) logic for strategic reasoning, named *Strategy Logic* [26] has been introduced and deeply studied. This logic has as a core engine the logic LTL. By simply considering as a core logic a prompt version of LTL [22], we get a prompt strategy logic for free. More involved, one can inject a prompt μ-calculus modal logic (instead of LTL) to have a proper prompt parity extension of Strategy Logic. Then, one can investigate opportune restrictions to the conceived logic to gain interesting complexities for the related decision problems. Overall, we recall that Strategy Logic is highly undecidable, while several of its interesting fragments are just to 2EXPTIME-COMPLETE. As another direction for future work, one may think to extend the prompt reasoning to infinite state systems by considering, for example, pushdown parity games [30, 3, 8]. However, this extension is rather than an easy task as one needs to rewrite completely the algorithms we have proposed.

References

[1] Almagor, S., Hirshfeld, Y., Kupferman, O.: Promptness in omega-Regular Automata, *ATVA'10*, LNCS 7388, 2010.

[2] Alur, R., Henzinger, T. A.: Finitary fairness, *ACM Trans. Program. Lang. Syst.*, **20**(6), 1998.

[3] Aminof, B., Legay, A., Murano, A., Serre, O., Vardi, M. Y.: Pushdown module checking with imperfect information, *Inf. Comput.*, **223**, 2013, 1–17.

[4] Aminof, B., Mogavero, F., Murano, A.: Synthesis of Hierarchical Systems., *FACS '11*, LNCS, Springer, 2011.

[5] Aminof, B., Mogavero, F., Murano, A.: Synthesis of hierarchical systems, *Science of Comp. Program.*, **83**, 2013, 56–79.

[6] B. Aminof and O. Kupferman and A. Murano: Improved model checking of hierarchical systems, *Inf. Comput.*, **210**, 2012, 68–86.

[7] Berwanger, D.: Admissibility in Infinite Games, *STACS'07*, 2007.

[8] Bouquet, A.-J., Serre, O., Walukiewicz, I.: Pushdown Games with Unboundedness and Regular Conditions, *FSTTCS*, 2003.

[9] Chatterjee, K., Doyen, L.: Energy parity games, *Theor. Comput. Sci.*, **458**, 2012, 49–60.

[10] Chatterjee, K., Doyen, L., Henzinger, T. A., Raskin, J.-F.: Generalized Mean-payoff and Energy Games, *FSTTCS'10*, LIPIcs 8, 2010.

[11] Chatterjee, K., Henzinger, T. A., Horn, F.: Finitary winning in ω-regular games, *ACM Trans. Comput. Logic*, **11**(1), 2009.

[12] Chatterjee, K., Henzinger, T. A., Jurdzinski, M.: Mean-Payoff Parity Games, *LICS'05*, 2005.

[13] Chatterjee, K., Jurdzinski, M., Henzinger, T. A.: Quantitative stochastic parity games, *SODA'04*, 2004.

[14] Clarke, E., Emerson, E.: Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic., *LP'81*, LNCS 131, 1981.

[15] Clarke, E., Grumberg, O., Peled, D.: *Model Checking.*, MIT Press, 2002.

[16] Emerson, E., Jutla, C.: Tree Automata, μ-Calculus and Determinacy, *FOCS'91*, 1991.

[17] Fijalkow, N., Zimmermann, M.: Cost-Parity and Cost-Streett Games, *FSTTCS'12*, 2012.

[18] Horn, F., Thomas, W., Wallmeier, N.: Optimal Strategy Synthesis in Request-Response Games, *ATVA'08*, LNCS 5311, 2008.

[19] Jurdzinski, M.: Deciding the Winner in Parity Games is in UP $\cap$ co-Up, *Inf. Process. Lett.*, **68**(3), 1998, 119–124.

[20] Kozen, D.: Results on the Propositional mu-Calculus., *TCS*, **27**(3), 1983, 333–354.

[21] Kupferman, O., Morgenstern, G., Murano, A.: Typeness for omega-regular Automata, *Int. J. Found. Comput. Sci.*, **17**(4), 2006, 869–884.

[22] Kupferman, O., Piterman, N., Vardi, M. Y.: From liveness to promptness, *Formal Methods in System Design*, **34**(2), 2009, 83–103.

[23] Kupferman, O., Vardi, M., Wolper, P.: An Automata Theoretic Approach to Branching-Time Model Checking., *JACM*, **47**(2), 2000, 312–360.

[24] Kupferman, O., Vardi, M., Wolper, P.: Module Checking, *IC*, **164**(2), 2001, 322–344.

[25] Kupferman, O., Vardi, M. Y.: Module Checking Revisited, *CAV'97*, 1254, 1997.

[26] Mogavero, F., Murano, A., Perelli, G., Vardi, M.: What Makes ATL* Decidable? A Decidable Fragment of Strategy Logic., *CONCUR'12*, LNCS 7454, Springer, 2012.

[27] Mogavero, F., Murano, A., Sorrentino, L.: On Promptness in Parity Games, *LPAR*, 2013.

[28] Pnueli, A.: The Temporal Logic of Programs., *FOCS'77*, 1977.

[29] Queille, J., Sifakis, J.: Specification and Verification of Concurrent Programs in Cesar., *SP'81*, LNCS 137, 1981.

[30] Walukiewicz, I.: Pushdown Processes: Games and Model-Checking, *Inf. Comput.*, **164**(2), 2001, 234–263.

[31] Zielonka, W.: Infinite Games on Finitely Coloured Graphs with Applications to Automata on Infinite Trees, *Theor. Comput. Sci.*, **200**(1-2), 1998, 135–183.