On Promptness in Parity Games*

Fabio Mogavero, Aniello Murano, and Loredana Sorrentino

Università degli Studi di Napoli Federico II

Abstract. *Parity games* are a powerful formalism for the automatic synthesis and verification of reactive systems. They are closely related to alternating ω-automata and emerge as a natural method for the solution of the μ-calculus model checking problem. 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 parity game variants 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 improve the complexity results of *cost* and *bounded-cost parity games*. Indeed, we provide solution algorithms showing that determining the winner of these games lies in UPTime ∩ CoUPTime.

1 Introduction

Parity games [13, 24] 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*, *discounted* payoff, *stochastic* and *multi-agent* games [6,7,9,10]. 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 turns 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

* 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.

often is even, otherwise, player $\forall$ wins the play. The problem of finding a winning strategy in parity games is in UPTIME $\cap$ COUPTIME [16] and the question whether or not a polynomial time solution exists is a long-standing open one.

In formal system design and verification [11, 12, 21, 23], parity games arise as a natural evaluation machinery for the automatic synthesis and verification of distributed and reactive systems [3–5, 20]. Specifically, in model-checking, one can check the correctness of a system with respect to a desired behavior, by checking whether a model of the system, that is, a *Kripke structure*, is correct with respect to a formal specification of its behavior, usually described in terms of a modal logic formula. In case the specification is given as a μ-calculus formula [17], the model checking question can be polynomially rephrased as a parity game [13].

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, it 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 [22] 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* [15]. 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, that is, the job must be printed in a reasonable time bound.

Lately, several works have focused on the above timing aspect in system specification. In [19], 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 [8], 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 [8] that the problem of determining the winner in a finitary parity game is in PTIME.

Recently, the work [8] has been generalized in [14] 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 [14], 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 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 condition. 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 [8] on classic arenas. We also refer to [8] 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), 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, since there is just one property to verify. 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 complexity 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 time complexity point of view. Obviously, this is different from what is done in [14] 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 [8], 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.

Due to the lack of space, proofs are omitted and reported in the full version.

2 Preliminaries

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

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 $\sigma_{\exists} \in \mathrm{Str}_{\exists} \subseteq \mathrm{Hst}_{\exists} \to \mathrm{Ps}$ (resp., $\sigma_{\forall} \in \mathrm{Str}_{\forall} \subseteq \mathrm{Hst}_{\forall} \to \mathrm{Ps}$) 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* $\mathsf{play} : \mathrm{Ps} \times (\mathrm{Str}_{\exists} \times \mathrm{Str}_{\forall}) \to \mathrm{Pth}$ 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 $\mathsf{play}(v, (\sigma_{\exists}, \sigma_{\forall}))$.

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 $\mathsf{pf} : \mathrm{Pth} \to \mathrm{Pf}$ 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 $\mathsf{pf} : \mathrm{Ps} \times (\mathrm{Str}_{\exists} \times \mathrm{Str}_{\forall}) \to \mathrm{Pf}$ 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.

Games. A *(extensive-form) game* is a tuple $\partial \triangleq \langle \widehat{\mathcal{A}}, \mathrm{Wn}, v_{\mathrm{o}} \rangle$, 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 ∂ 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., $\mathsf{pf}(\sigma_{\exists}, \sigma_{\forall}) \notin \mathrm{Wn}$).

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).

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* with 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 with respect to 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, that is, some of them have to be satisfied almost everywhere and the remaining ones, over all the play. We

	Non-Prompt	Prompt
Non-Full	Parity (P)	Prompt Parity (PP) ≡ 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.

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. Note that, 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).

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 $\Game \triangleq \langle \widehat{\mathcal{A}}, \mathrm{Wn}, v_0 \rangle$ be a game, where the payoff arena $\widehat{\mathcal{A}}$ is induced by a colored arena $\widetilde{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl} \rangle$.

Parity condition (P) $\Game$ 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).

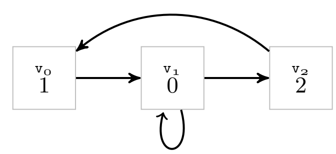

Fig. 1. Colored Arena $\widetilde{\mathcal{A}_1}$.

Full Parity condition (FP) $\Game$ 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 $\mathrm{Rq}(c) = \mathrm{Rs}(c)$ *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.

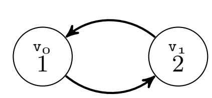

Fig. 2. Colored Arena $\widetilde{\mathcal{A}_2}$.

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 $\eth \triangleq \langle \widehat{\mathcal{A}}, \mathrm{Wn}, v_0 \rangle$ be a game, where the payoff arena $\widehat{\mathcal{A}}$ is induced by a (colored) weighted arena $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$.

Prompt Parity condition (PP) $\eth$ 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 $c_\pi \triangleq \mathsf{pf}(\pi) = (c, w)$, where $c = 3 \cdot (1 \cdot 2)^\omega$ and $w = 2 \cdot (1 \cdot 0)^\omega$, and set of requests $\mathrm{Rq}(c) = \{0\} \cup \{2n + 1 : n \in \mathrm{N}\}$. Then, $c_\pi \in \mathrm{Wn}$, since the prompt parity condition is satisfied by choosing $\mathrm{R} = \{0\}$ and $b = 1$.

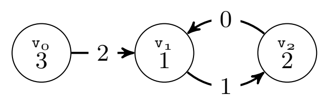

Fig. 3. Weighted Arena $\overline{\mathcal{A}_3}$.

Full-Prompt Parity condition (FPP) $\eth$ 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 $c_\pi \triangleq \mathsf{pf}(\pi) = (c, w)$ be its payoff, with $c \in 3 \cdot 4 \cdot ((3 \cdot 4)^* \cdot (1 \cdot 4)^*)^\omega$ and $w \in 2 \cdot ((0 \cdot 2)^* \cdot (0 \cdot 1)^*)^\omega$. Then, $c_\pi \in \mathrm{Wn}$, since the full-prompt parity condition is satisfied as all requests are responded by color 4 with a delay bound $b = 2$.

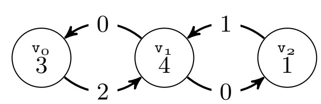

Fig. 4. Weighted Arena $\overline{\mathcal{A}_4}$.

Remark 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 above two cases just analyzed correspond to the finitary parity and bounded parity conditions studied in [8].

Cost Parity condition (CP) [14] $\eth$ 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

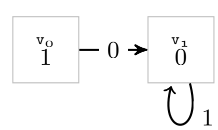

Fig. 5. Weighted Arena $\overline{\mathcal{A}_5}$.

requests, except at most a finite number of them (possibly different from the previous ones) 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 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 $c_\pi \triangleq \mathsf{pf}(\pi) = (c, w)$, where $c = 1 \cdot 0^{\omega}$ and $w = 0 \cdot 1^{\omega}$, and set of requests $\mathrm{Rq}(c) = \{0\}$. Then, $c_\pi \in \mathrm{Wn}$, since the prompt parity condition is satisfied with $\mathrm{R} = \mathrm{R}' = \{0\}$ and $b = 0$.

Bounded-Cost Parity condition (BCP) [14] $\Game$ 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 $c_\pi \triangleq \mathsf{pf}(\pi) = (c, w)$, where $c = 1 \cdot 0^{\omega}$, and set of requests $\mathrm{Rq}(c) = \{0\}$. Then, $c_\pi \in \mathrm{Wn}$, since the prompt parity condition is satisfied with $\mathrm{R} = \{0\}$ and $b = 1$.

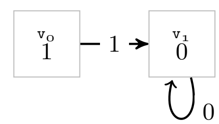

Fig. 6. Weighted Arena $\overline{\mathcal{A}_6}$.

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

We now study the relationships among all parity conditions given above.

4.1 Positive Relationships

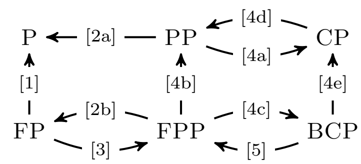

Fig. 7. Implication Schema.

We now prove all positive relationships among the given conditions and report them in Figure 7, where an arrow from a condition Wn_1 to another one Wn_2 means that the former implies the latter. Namely, if player $\exists$ wins a game under Wn_1 condition, then it also wins the game under the one Wn_2, over the same arena. The label on the edges indicates next theorem's item 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. Observe that, in the following, we refer to $\widehat{\mathcal{A}}$, $\widetilde{\mathcal{A}}$, $\overline{\mathcal{A}}$ indicating, respectively the payoff, colored and weighted arenas.

Theorem 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 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 an 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 an arena* $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$ *and its underlying arena* $\widetilde{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl} \rangle$ *and one among (a)* $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{PP}, \mathrm{P})$ *and (b)* $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{FPP}, \mathrm{FP})$ *hold.*
3. $\widehat{\mathcal{A}_2}$ *and* $\widehat{\mathcal{A}_1}$ *are finite and induced, respectively, by an arena* $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$ *and its underlying 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 an arena* $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$ *and one among (a)* $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{PP}, \mathrm{CP})$*, (b)* $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{FPP}, \mathrm{PP})$*, (c)* $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{FPP}, \mathrm{BCP})$*, (d)* $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{CP}, \mathrm{PP})$*, (e)* $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{BCP}, \mathrm{CP})$ *hold.*
5. $\widehat{\mathcal{A}_1} = \widehat{\mathcal{A}_2}$ *are induced by an 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* $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{BCP}, \mathrm{FPP})$*.*

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 1. *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 arenas* $\widehat{\mathcal{A}_{\mathrm{FPP}}}$ *and* $\widehat{\mathcal{A}_{\mathrm{FP}}}$ *induced, respectively, by an arena* $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$ *and its underlying 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 2. *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 arena* $\widehat{\mathcal{A}}$ *induced by an 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 3. *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 arena* $\widehat{\mathcal{A}}$ *induced by an 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

We, now, show a list of counterexamples to point out that some winning conditions are not equivalent to other ones and report the corresponding results in Figure 8, where an arrow from a condition Wn_1 to another condition Wn_2 means that there is 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 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 listed ones together with the implication results of Theorem 1: (P, FPP), (P, CP), (P, BCP), (FP, FPP), (FP, CP), (FP, BCP), (PP, FPP), (CP, FP), (CP, FPP), (CP, BCP), and (BCP, FPP).

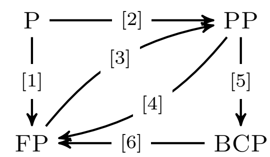

Fig. 8. Counterexample Schema.

Theorem 2. *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 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 an 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 an arena* $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$ *and its underlying 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* $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$ *and its underlying 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 an arena* $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$ *and its underlying 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* $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$ *and* $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{PP}, \mathrm{BCP})$*;*
6. $\widehat{\mathcal{A}_1}$ *and* $\widehat{\mathcal{A}_2}$ *are induced, resp., by* $\overline{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl}, \mathrm{Wg}, \mathsf{wg} \rangle$*, with* $\mathsf{wg}(v) = 0$*, for* $v \in \mathrm{Ps}$*, and its underlying arena* $\widetilde{\mathcal{A}} = \langle \mathcal{A}, \mathrm{Cl}, \mathsf{cl} \rangle$ *and* $(\mathrm{Wn}_1, \mathrm{Wn}_2) = (\mathrm{BCP}, \mathrm{FP})$.

5 Polynomial Reductions

In this section, we face the computational complexity of solving FP, PP, and BCP games. Then, due to the relationships among the winning conditions described above, we extend 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 built game encapsulates in 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. A transition table is an automaton without acceptance condition. It is used to represent the information of the winning condition mentioned above. Then, the product operation allows to pass this information to 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 $\perp < i$, for all $i \in \mathrm{Cl}$. Intuitively, $\perp$ 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 $\mathrm{R}_\perp \triangleq \{\perp\} \cup \mathrm{R}$.

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. This 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 $\mathcal{T}$. Intuitively, $\mathcal{T}$ keeps track, along every play, 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, $\mathcal{T}$ uses as states $\perp$ and all possible request values, and its transition function is defined as follows: if a request is satisfied, then $\mathcal{T}$ 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 $\mathcal{A}^\star$ built as the product of the original arena with $\mathcal{T}$ and use as colors the values 1 and 2, assigned as follows: if a position contains $\perp$, color it with 2, otherwise, color it with 1. By definition of full parity and

Büchi games, we have that a Büchi game is won over $\mathcal{A}^\star$ if and only if the full parity game is won over the original arena. Indeed, over a play of $\mathcal{A}^\star$, meeting $\perp$ infinitely often means that all requests found over the corresponding play of the old arena are satisfied. The formal construction of $\mathcal{T}$ and the $\mathcal{A}^\star$ follow. For a given FP game $\partial \triangleq \langle \widehat{\mathcal{A}}, \mathrm{FP}, v_o \rangle$ 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 $\partial^\star = \langle \widehat{\mathcal{A}^\star}, \mathrm{B}, (v_o, \perp) \rangle$ induced by $\widetilde{\mathcal{A}^\star}$ is such that player $\exists$ wins ∂ iff it wins $\partial^\star$.

Theorem 3. *For every* FP *game* ∂ *with* $k \in \mathbb{N}$ *priorities, there is a* B *game* $\partial^\star$*, with order* $|\partial^\star| = \mathrm{O}(|\partial| \cdot k)$*, such that player* $\exists$ *wins* ∂ *iff it wins* $\partial^\star$*.*

5.3 From Bounded-Cost Parity to Parity

We now 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 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 $\perp$ and $(\perp, 0)$ indicates 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 $(\perp, 0)$, 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 $\supsetneq \triangleq \langle \widehat{\mathcal{A}}, \mathrm{BCP}, v_{\mathrm{o}} \rangle$ 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 $\mathrm{St} \triangleq \{\circledast\} \cup \mathrm{R}_{\perp} \times [0, s]$, 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,

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

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

Theorem 4. *For every* BCP *game* $\supsetneq$ *with* $k \in \mathbb{N}$ *priorities and sum of weights* $s \in \mathbb{N}$*, there is a* P *game* $\supsetneq^{\star}$*, with order* $|\supsetneq^{\star}| = \mathrm{O}(|\supsetneq| \cdot k \cdot s)$*, such that player* $\exists$ *wins* $\supsetneq$ *iff it wins* $\supsetneq^{\star}$*.*

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 and it is used to later check the guess states. As first state we have the tuple $((\perp, 0, D), 0))$ 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, then it goes and remains forever in state $\circledast$; if the request is satisfied then it goes to $((\perp, d, D), 0)$; 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 is the one present in the current state plus the weight of the traversed edge. The transition function over an existential state is defined as follows. If d is equal to the maximum allowable number of positions having an odd priority (h), then the computation remains in the same (deterministic) state; otherwise, the computation moves to a state in which the second component is incremented by 1. Note that the guess part is similar to that one performed to translate a nonderministic co-Büchi automaton into a Büchi one [18]. Finally, we color the obtained arena as we did for the above BCP case. 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 $\perp$. 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 $\Game \triangleq \langle \widehat{\mathcal{A}}, \mathrm{PP}, v_{\mathrm{o}} \rangle$ 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 $\mathrm{Z}_\alpha \triangleq \mathrm{R}_\perp \times [0, h] \times \alpha$) and its transition function defined as follows: $\mathsf{tr}(\divideontimes, (c, w)) \triangleq \divideontimes$ and, additionally:

- $\mathsf{tr}(((r, d, D), k), (c, w)) \triangleq \begin{cases} ((\perp, d, D), 0), & \text{if } r < c \wedge c \equiv 0 \pmod 2; \\ \divideontimes, & \text{if } k + w > s; \\ ((\max\{r, c\}, d, \exists), k + w), & \text{otherwise.} \end{cases}$
- $\mathsf{tr}(((r, d, \exists), k)) \triangleq \begin{cases} \{((r, d, D), k)\}, & \text{if } d = h; \\ \{((r, d, D), k), ((\perp, d + 1, D), 0)\}, & \text{otherwise.} \end{cases}$

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 $\Game^\star = \langle \widehat{\mathcal{A}^\star}, \mathrm{P}, (v_{\mathrm{o}}, ((\perp, 0, D), 0)) \rangle$ induced by $\widetilde{\mathcal{A}^\star}$ is such that player $\exists$ wins $\Game$ iff it wins $\Game^\star$.

Theorem 5. *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* $\Game^\star$*.*

Observe that the estimation on the size of $\Game^\star$ is quite coarse since several type of states can not be reached by the initial position.

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 $t = ((\perp, d, D), 0)$ for some $d \in [0, h]$ then $\mathsf{cl}^\star((v, t)) = 2$ else $\mathsf{cl}^\star((v, t)) = 1$. Then, the B game $\Game^\star = \langle \widehat{\mathcal{A}^\star}, \mathrm{B}, (v_{\mathrm{o}}, ((\perp, 0, D), 0)) \rangle$ induced by $\widetilde{\mathcal{A}^\star}$ is such that player $\exists$ wins $\Game$ iff it wins $\Game^\star$.

Theorem 6. *For every* PP *game* $\supseteq$ *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* $\supseteq^{\star}$*, with order* $|\supseteq^{\star}| = \mathrm{O}(|\supseteq|^2 \cdot k \cdot s)$*, such that player* $\exists$ *wins* $\supseteq$ *iff it wins* $\supseteq^{\star}$*.*

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.

As games already addressed in literature, we studied cost parity and bounded-cost parity and, for both of them, we provided algorithms that improve their known complexity. As new parity games, we investigated full parity, full-prompt parity, and prompt parity. We showed that full parity is in PTime, prompt parity and cost parity are equivalent and both in UPTime ∩ CoUPTime. The latter improves the known complexity result to solve cost parity games because our algorithm reduce the original problem to a unique parity game while their one performs "several calls" to a parity games 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.

Conditions	Colored Arena	(Colored) Weighted arena
Parity (P)	UPTime ∩ CoUPTime [16]	$\hookleftarrow$
Full Parity (FP)	PTime [Thm 3]	$\hookleftarrow$
Prompt Parity (PP)	PTime [Thm 6]	UPTime ∩ CoUPTime [Thm 5]
Full Prompt Parity (FPP)	$\hookleftarrow$	PTime [FP + Cor 1]
Cost Parity (CP)	PTime [PP + Cor 2]	UPTime ∩ CoUPTime [PP + Cor 2]
Bounded Cost Parity (BCP)	PTime [FPP + Cor 3]	UPTime ∩ CoUPTime [Thm 4]

Table 3. Summary of all winning condition complexities.

References

1. S. Almagor, Y. Hirshfeld, and O. Kupferman. Promptness in omega-Regular Automata. In *ATVA'10*, LNCS 7388, pages 22–36, 2010.
2. R. Alur and T. A. Henzinger. Finitary fairness. *ACM Trans. Program. Lang. Syst.*, 20(6), 1998.
3. B. Aminof, F. Mogavero, and A. Murano. Synthesis of hierarchical systems. In *FACS '11*, LNCS, pages 42–60. Springer, 2011.
4. B. Aminof, F. Mogavero, and A. Murano. Synthesis of hierarchical systems. *Science of Comp. Program.*, 2013. DOI: http://dx.doi.org/10.1016/j.scico.2013.07.001.
5. B. Aminof and O. Kupferman and A. Murano. Improved model checking of hierarchical systems. *Inf. Comput.*, 210:68–86, 2012.
6. D. Berwanger. Admissibility in infinite games. In *STACS'07*, pages 188–199, 2007.
7. K. Chatterjee, L. Doyen, T. A. Henzinger, and J.-F. Raskin. Generalized mean-payoff and energy games. In *FSTTCS'10*, LIPIcs 8, pages 505–516, 2010.
8. K. Chatterjee, T. A. Henzinger, and F. Horn. Finitary winning in ω-regular games. *ACM Trans. Comput. Logic*, 11(1), 2009.
9. K. Chatterjee, T. A. Henzinger, and M. Jurdzinski. Mean-payoff parity games. In *LICS'05*, pages 178–187, 2005.
10. K. Chatterjee, M. Jurdzinski, and T. A. Henzinger. Quantitative stochastic parity games. In *SODA'04*, pages 121–130, 2004.
11. E.M. Clarke and E.A. Emerson. Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic. In *LP'81*, LNCS 131, pages 52–71, 1981.
12. E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking.* MIT Press, 2002.
13. E.A. Emerson and C. Jutla. Tree automata, μ-calculus and determinacy. In *FOCS'91*, pages 368–377, 1991.
14. N. Fijalkow and M. Zimmermann. Cost-parity and cost-streett games. In *FSTTCS'12*, pages 124–135, 2012.
15. F. Horn, W. Thomas, and N. Wallmeier. Optimal strategy synthesis in request-response games. In *ATVA'08*, LNCS 5311, pages 361–373, 2008.
16. M. Jurdzinski. Deciding the winner in parity games is in up $\cap$ co-up. *Inf. Process. Lett.*, 68(3):119–124, 1998.
17. D. Kozen. Results on the Propositional mu-Calculus. *TCS*, 27(3):333–354, 1983.
18. O. Kupferman, G. Morgenstern, and A. Murano. Typeness for omega-regular automata. *Int. J. Found. Comput. Sci.*, 17(4):869–884, 2006.
19. O. Kupferman, N. Piterman, and M. Y. Vardi. From liveness to promptness. *Formal Methods in System Design*, 34(2):83–103, 2009.
20. O. Kupferman, M. Vardi, and P. Wolper. Module Checking. *IC*,164(2):322–344, 2001.
21. O. Kupferman, M.Y. Vardi, and P. Wolper. An Automata Theoretic Approach to Branching-Time Model Checking. *JACM*, 47(2):312–360, 2000.
22. A. Pnueli. The Temporal Logic of Programs. In *FOCS'77*, pages 46–57, 1977.
23. J.P. Queille and J. Sifakis. Specification and Verification of Concurrent Programs in Cesar. In *SP'81*, LNCS 137, pages 337–351, 1981.
24. W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theor. Comput. Sci.*, 200(1-2):135–183, 1998.