

Hierarchical Cost-Parity Games

Laura Bozzelli¹, Aniello Murano^{*2}, Giuseppe Perelli^{†3}, and Loredana Sorrentino⁴

1 Università degli Studi di Napoli Federico II, Naples, Italy

2 Università degli Studi di Napoli Federico II, Naples, Italy

3 University of Oxford, Oxford, UK

4 Università degli Studi di Napoli Federico II, Naples, Italy

Abstract

Cost-parity games are a fundamental tool in system design for the analysis of reactive and distributed systems that recently have received a lot of attention from the formal methods research community. They allow to reason about the time delay on the requests granted by systems, with a bounded consumption of resources, in their executions.

In this paper, we contribute to research on Cost-parity games by combining them with *hierarchical systems*, a successful method for the succinct representation of models. We show that determining the winner of a *Hierarchical Cost-parity Game* is PSPACE-COMplete, thus matching the complexity of the proper special case of *Hierarchical Parity Games*. This shows that reasoning about temporal delay can be addressed at a free cost in terms of complexity.

1998 ACM Subject Classification F.3.1 Specifying and Verifying and Reasoning about Programs

Keywords and phrases Parity Games, Cost-Parity Games, Hierarchical Systems, System Verification

Digital Object Identifier 10.4230/LIPIcs.TIME.2017.6

1 Introduction

In formal system design and verification [11, 12, 20, 26], *Parity Games* represent a fundamental machinery for the automatic synthesis and verification of concurrent and reactive systems [5, 6, 7, 21, 22]. The determinacy and the memorylessness of parity games is crucial in various theoretical areas useful in formal verification, among which we mention automata theory, temporal and modal logics, and monadic second-order logics. For instance, the emptiness problem of alternating tree automata [14] as well as model checking and satisfiability in modal μ -calculus [18] can be reduced to deciding the winner of a parity game. In particular, model checking μ -calculus is equivalent via linear time reduction to this problem [13].

As pointed out in [15, 23, 24], the parity winning condition corresponds to a qualitative *request-response condition* [17]: Player 0 wins a play of infinite duration if all but finitely many odd colors (which we think of as requests) are followed by larger even colors (which we think of as responses). In this setting, there is no bound on the wait time, *i.e.*, the number of steps that elapse between a request and its first response in the play. On the other hand, in many applications, it is important to bound the wait time. In the last decade, many papers have focused on quantitative aspects, in particular boundedness requirements, of

* The author is partially supported by the INdAM research project 2017 “Logica e Automi per il Model Checking”.

† The author acknowledges with gratitude the the financial support of the ERC Advanced Investigator grant 291528 (“RACE”) at Oxford.



formal verification [1, 19, 10], including parity games [10, 15, 23, 24]. In [19], the authors introduce Prompt LTL, an extension of standard LTL [25] with the prompt-eventually operator F_p : a finite system satisfies a Prompt LTL formula φ iff there is a bound on the wait time for all the prompt-eventually subformulas of φ in all the computations of the system. The automata-theoretic counterpart of the F_p operator has been investigated in [1]. Parity games extended with promptness requirements, the so-called finitary parity games, have been studied in [10]. The finitary parity condition [10] extends the parity condition by additionally requiring the existence (along the given play) of a bound k such that almost every odd color is answered within at most k steps. Surprisingly, finitary parity games are solvable in polynomial time, and thus simpler than parity games (according to the state-of-the-art). A meaningful generalization of finitary games is represented by the class of parity games with costs [15] (in the following, referred as cost-parity games). In such games, transitions are labeled by non-negative integers (costs). The cost of traversing a transition can be used to model resource consumption. The goal of Player 0 consists then in ensuring the underlying parity condition by using bounded resources: a play is winning for Player 0 if there is a bound k such that almost every odd color is followed by a larger even color that is reached with cost at most k . On the other hand, Player 1's goal is to exhaust the resources by making the cost unbounded. Note that Player 1's objective is not an ω -regular property, and in general, Player 1 needs infinite memory to win such games. However, cost-parity games enjoy some nice properties: Player 0 has memoryless winning strategies and determining the winner lies in $\text{NP} \cap \text{coNP}$. This upper bound has been recently improved to $\text{UP} \cap \text{coUP}$ in [24], proving thus that the increased expressiveness with respect to parity conditions comes at a free cost in terms of complexity.

In the recent years, many other quantitative extensions of parity games have been introduced. Among them we would like to mention Mean-Payoff Parity Games [9], whose winning condition is a combination of a parity and a mean-payoff objective, and Energy Parity Games [8]. These last ones are played over weighted arenas, and the winning condition extends the parity condition by additionally requiring that the sum of the weights along a play (interpreted as level of energy, or resource usage) remains always positive.

A well-known issue in formal verification is that the translation of a high-level description of a system into a formal model, typically given by a finite-state machine (FSM), often involves an exponential blow-up in the size of the FSM, thus affecting the efficiency of the analysis procedures both in theory and practice. Several sources of this blow-up have been identified in the literature. A well-studied one is the ability of components in the system to work in parallel and communicating with each other, possibly using variables. The impact of the concurrent setting on analysis problems is well-known: it costs an exponential, leading to the so called state-explosion problem. Another source of the blow-up in the translation of systems into FSMs is that in high-level sequential programming, one can specify components only once and then can reuse them in different contexts, leading to modularity and succinct system representation. A smart way to represent such modularity is by means of *hierarchical FSM*, where some of the states of the FSM are boxes (superstates) which correspond to nested FSMs (the reused components). The naive approach to model checking such systems is to 'flatten' them by repeatedly substituting references to substructures with copies of them. This results in a flat FSM whose size is exponential in the nesting depth of the hierarchical system. However, differently from the concurrent setting, a wiser approach avoiding flattening, for the case of model checking against temporal logics like LTL, CTL and the more expressive modal μ -calculus, is beneficial in terms of complexity [3, 4, 5, 16]. Parity games have also been investigated under the hierarchical

setting. In [5], Aminof et al. prove that deciding the winner in a *Hierarchical Parity Game* (*HPG*) is a PSPACE-complete problem. The technique used in [5] is based on the observation that even though a sub-arena may appear in different contexts, it is possible to extract information about the sub-arena that is independent of the context in which it appears.

In this paper, we further investigate the power of hierarchical representation by introducing and studying *Cost-parity Games over Hierarchical Systems* (*HCPG*). As main result, we establish that the problem of solving *HCPG* is PSPACE-complete, which matches the complexity of the proper special case of hierarchical parity games (*HPG*). The proposed approach for solving the considered problem generalizes in a non-trivial and sophisticated manner the one exploited in [5] for solving *HPG*, and is based on the notion of summary function for a memoryless strategy σ of Player 0 in a given sub-arena. Such a function records in a finite and efficient way the overall behavior of all the finite plays of σ leading to exit states of the sub-arena with respect to requests and responses, by finitely abstracting the set of associated costs and delays. The algorithm for solving *HCPG* then solves a sequence of flat cost-parity games obtained by replacing sub-arenas by simple gadgets (depending only on the set of colors and exit states of the sub-arena) that implement the summary functions.

The sequel of the paper is structured as follows. In Section 2, we first recall the framework of cost-parity games. Then, we introduce hierarchical cost-parity games and describe our solution approach in Section 3. Finally, we give few conclusions and future work directions in Section 4. Due to space constraints, some proofs are omitted.

2 Preliminaries

Let \mathbb{N} be the set of natural numbers. For all $i, j \in \mathbb{N}$, with $i \leq j$, $[i, j]$ denotes the set of natural numbers h such that $i \leq h \leq j$. We fix a non-empty finite set C of natural numbers of the form $[0, j]$ for some $j \in \mathbb{N}$, which represents the set of colors for the given cost-parity winning condition. We denote by C_e and C_o the sets of even and odd colors in C , respectively. We assume that the maximal color j in C , denoted by C_o^{max} , is odd.

For an alphabet Σ , and a non-empty finite or infinite word w over Σ , we denote by $|w|$ the length of w (we set $|w| = \infty$ if w is infinite). Moreover, for all $i, j \geq 1$, with $i \leq j$, $w(i)$ is the i -th letter of w , while $w[i, j]$ denotes the finite subword of w given by $w(i) \cdots w(j)$, and w^i the prefix of w from position i , i.e., the word $w(i)w(i+1) \dots$.

2.1 Cost-Parity Games

We recall the framework of Cost-parity games [15] which are two-player turn-based games played on finite graphs equipped with a Cost-parity winning condition. In such a setting, Player 0 wins a play of infinite duration if there is a bound $\ell \in \mathbb{N}$ such that almost all odd colors (which we think of as requests) are followed by larger even colors (which we think of as responses) that are reached with cost at most ℓ .

A state-transition graph or FSM is a tuple $\langle S, R, in \rangle$ consisting of a finite set S of states, a transition relation $R \subseteq S \times S$, and an initial state $in \in S$. For a state $s \in S$, we write $R(s) = \{s' \in S \mid (s, s') \in R\}$ for the set of successors of s . A path in the FSM is a non-empty finite or infinite word π over S such that $\pi(i+1) \in R(\pi(i))$ for all $i \in [1, |\pi| - 1]$.

An *arena* is a tuple $\mathcal{A} = \langle S, S_0, S_1, R, in \rangle$ consisting of an FSM $\langle S, R, in \rangle$ and a partition $\{S_0, S_1\}$ of S into the states of Player 0 (drawn as circles) and the states of Player 1 (drawn as rectangles). A play of a game over \mathcal{A} proceeds by moving a token on the states of \mathcal{A} , starting at some state. If the token is placed on a state $s \in S_0$ (resp., $s \in S_1$), then the play ends if s has no successors (we call such a state a *terminal state*); otherwise, Player 0 (resp.,

Player 1) chooses a successor s' of s and moves the token to s' . Formally, a *play* of \mathcal{A} is a *maximal* path of \mathcal{A} , i.e., a path π in the underlying FSM such that either π is infinite, or π is finite and ends at a terminal state.

Let $p \in \{0, 1\}$ and S_p^N be the set of non-terminal states of Player p . A *strategy* for Player p is a mapping $\sigma : S^* \cdot S_p^N \mapsto S$ assigning to each non-empty sequence of states $w \cdot s \in S^* \cdot S_p^N$ leading to a non-terminal state s of Player p , a successor of s . A play π is consistent with the strategy σ if for all $k \in [1, |\pi| - 1]$ such that $\pi(k) \in S_p^N$, it holds that $\pi(k+1) = \sigma(\pi[1, k])$. The strategy σ is *memoryless* if its output does not depend on the whole prefix of the play, but only on the last position, i.e., if for all $w \cdot s \in S^* \cdot S_p^N$, $\sigma(w \cdot s) = \sigma(s)$. We can thus represent a memoryless strategy as a mapping $\sigma : S_p^N \rightarrow S$.

A (zero-sum) game is a pair $\langle \mathcal{A}, \text{Win} \rangle$ consisting of an arena $\mathcal{A} = \langle S, S_0, S_1, R, \text{in} \rangle$ and a subset Win of infinite plays which are winning for Player 0. An infinite play π is *winning for Player 1* if it is not winning for Player 0. A *finite* play π is winning for Player p if π ends at a state of the opponent Player $1 - p$. A strategy σ for Player p is *winning from a state s* if all the plays π starting from s which are consistent with the strategy σ are winning for Player p . In such a case, we say that state s is winning for Player p . A game is determined if for each state s , s is winning for one of the players. Note that since for all strategies σ^0 and σ^1 of Player 0 and Player 1, respectively, there is a unique play starting from s which is consistent with both σ^0 and σ^1 , in (zero-sum) games, a state s cannot be winning for both the players. Solving a game consists in checking whether the initial state is winning for Player 0.

Cost-parity winning conditions

We, now, recall the class of Cost-parity winning conditions. A *Cost-parity arena* $\mathcal{G} = \langle \mathcal{A}, \text{Cost}, \Omega \rangle$ over the set C of colors consists of an arena $\mathcal{A} = \langle S, S_0, S_1, R, \text{in} \rangle$, a transition-labeling $\text{Cost} : R \mapsto \{0, 1\}$ (cost function), and a coloring mapping $\Omega : S \mapsto C$ assigning to each state a color in C . Note that according to [15], the definition of transition-labelling only allows cost 0 or 1 on a transition. Having arbitrary costs in \mathbb{N} would not change our results, as we are interested in boundedness questions only. We extend the transition-labeling to a cost function Cost over paths π obtained by counting the number of increment transitions (i.e., 1-labeled transitions) traversed along the path, i.e., $\text{Cost}(\pi) = \sum_{i=2}^{|\pi|} \text{Cost}(\pi(i-1), \pi(i))$. Note that $\text{Cost}(\pi) \in \mathbb{N} \cup \{\infty\}$.

The pair (Cost, Ω) induces a winning condition for Player 0, where the occurrence of an odd color along a play π is interpreted as a *request*, for which there has to be a *response* later on the play by a higher even color. Formally, let π be a finite or infinite path of \mathcal{A} . A *request in π* is a position k along π such that $\pi(k)$ has *odd* color. For an odd color c , a c -request in π is a request k in π such that $\Omega(\pi(k)) = c$. Moreover, we define $\text{Ans}(c) = \{c' \in C_e \mid c' \geq c\}$, i.e., the set of even colors that answers a request of color c . For a request k in π , let r_k be the smallest position $k' \geq k$ that answers to request k , i.e., such that $\Omega(\pi(k')) \in \text{Ans}(\Omega(\pi(k)))$, if such positions k' exist, and let $r_k = |\pi|$ otherwise. In the first (resp., second) case, we say that the request k is *answered* (resp., *unanswered*) in π . The *delay* of the request k in π , denoted by $\text{dl}(\pi, k)$, then is defined as the cost of the infix of π from the request k to position r_k , i.e., $\text{Cost}(\pi[k, r_k])$ if $r_k \neq \infty$, and $\text{Cost}(\pi^k)$ otherwise. The *cost-parity* winning condition induced by (Cost, Ω) , written $\text{CostParity}(\text{Cost}, \Omega)$, is then the set of infinite plays π such that there is $n \geq 1$ and a bound $\ell \in \mathbb{N}$ so that for all requests k in π with $k \geq n$, $\text{dl}(\pi, k) \leq \ell$ and the request k is answered in π . Thus, an infinite play $\pi \in \text{CostParity}(\text{Cost}, \Omega)$ iff there is bound ℓ such that all but finitely many requests are answered with cost less than ℓ . Note that $\text{CostParity}(\text{Cost}, \Omega)$ is *prefix-independent*, i.e., for all infinite plays π and $k \geq 1$, $\pi \in \text{CostParity}(\text{Cost}, \Omega)$ iff $\pi^k \in \text{CostParity}(\text{Cost}, \Omega)$. We recall the following known result.

► **Theorem 1** ([15]). *Cost-parity games are determined and Player 0 has memoryless winning strategies from the winning Player 0 states. Moreover, solving a cost-parity game $\mathcal{G} = \langle \mathcal{A}, \text{Cost}, \Omega \rangle$ with k colors can be done in time $|\mathcal{A}|^{O(k \cdot \log k)}$ and in polynomial space.*

For technical convenience, we also consider a generalization of cost-parity arenas, called *partial cost-parity arenas*, where one considers as additional input a subset *Exit* of the set of terminal states, called *exit states*. Finite plays ending at states in *Exit* are assumed to be *non-winning* for either player and have an undefined value. In this setting, a *non-losing strategy for Player p from state s* is a strategy σ for Player p such that each play starting from s which is consistent with σ and does not lead to an exit state is winning for Player p . A *non-losing strategy* is a non-losing strategy for Player 0 from the initial state in . For a strategy σ for Player 0, an *exit play* of σ is a finite play starting from in and ending at an exit state which is consistent with σ . For $s \in \text{Exit}$, an *s-exit play* of σ is an exit play of σ leading to s . Two partial cost-parity arenas $\mathcal{G} = \langle \mathcal{A}, \text{Cost}, \Omega, \text{Exit} \rangle$ and $\mathcal{G}' = \langle \mathcal{A}', \text{Cost}', \Omega', \text{Exit}' \rangle$ have the same interface if $\text{Exit} = \text{Exit}'$, \mathcal{G} and \mathcal{G}' have the same initial state in , and for each $s \in \{in\} \cup \text{Exit}$, the colors and the players of state s in \mathcal{G} and \mathcal{G}' coincide.

2.2 Hierarchical Cost-Parity Games

A *Hierarchical Cost-Parity Game* is a cost-parity game played over a (flat) arena induced by a *hierarchical arena*. The latter is a standard hierarchical FSM [4] in which the set of nodes of each of the underlying FSMs is partitioned into nodes belonging to Player 0 and nodes belonging to Player 1. We refer to the underlying FSMs as modular sub-arenas. Formally, a hierarchical arena is a tuple $\mathcal{V} = \langle \mathcal{V}_1, \dots, \mathcal{V}_n \rangle$ of modular sub-arenas, where each \mathcal{V}_i is in turn a tuple of the form $\langle N_i, N_i^0, N_i^1, B_i, in_i, Exit_i, Y_i, E_i \rangle$ consisting of the following components:

- A finite set N_i of nodes which is partitioned into a set N_i^0 of nodes of Player 0 and a set N_i^1 of nodes of Player 1, and a finite set B_i of *boxes*. We assume that $N_1, \dots, N_n, B_1, \dots, B_n$ are pairwise disjoint.
- An initial node or entry $in_i \in N_i$,¹ and a subset $Exit_i$ of N_i called *exit-nodes*. We assume that $Exit_1 = \emptyset$, i.e., the top-level sub-arena \mathcal{V}_1 has no exits.
- An indexing function $Y_i : B_i \rightarrow \{i+1, \dots, n\}$ that maps each box b of \mathcal{V}_i to an index $Y_i(b) > i$. The box b represents a reference to the definition of the sub-arena $\mathcal{V}_{Y_i(b)}$.
- An edge relation E_i . Each edge in E_i is a pair (u, v) such that: (i) the source u is either a node of \mathcal{V}_i or a pair (b, e) , where b is a box of \mathcal{V}_i and e is an exit-node of the sub-arena that b refers to, and (ii) the target v is either a node or a box of \mathcal{V}_i .

Define $N = \bigcup_{i=1}^n N_i$ (the set of \mathcal{V} -nodes), $E = \bigcup_{i=1}^n E_i$ (the set of \mathcal{V} -edges), and $Exit = \bigcup_{i=1}^n Exit_i$ (the set of \mathcal{V} -exit-nodes). In a modular sub-arena, the edges connect nodes and boxes with one another. Edges entering a box implicitly lead to the unique entry-node of the sub-arena that the box refers to. On the other hand, an edge exiting a box needs to explicitly specify the identity of the exit-node among the possible exit-nodes of the sub-arena associated with that box. The *size* $|\mathcal{V}_i|$ of a modular sub-arena \mathcal{V}_i is $|N_i| + |B_i| + |E_i|$. The size $|\mathcal{V}|$ of \mathcal{V} is $\sum_{i=1}^n |\mathcal{V}_i|$. The *nesting depth* of \mathcal{V} is the length of the longest chain i_1, i_2, \dots, i_j of indices in $[1, n]$ such that a box of \mathcal{V}_{i_l} is mapped to i_{l+1} for all $l \in [1, j-1]$. Note that the fact that boxes of a sub-arena can only refer to sub-arenas with a greater index implies that the nesting depth of \mathcal{V} is finite. Such a restriction does not exist in the *recursive* setting [2].

¹ We assume a single entry for each sub-arena. Multiple entries can be handled by duplicating sub-arenas.

6:6 Hierarchical Cost-Parity Games

A *Hierarchical Cost-Parity Arena (HCPA)*, for short) over C is a tuple $\mathcal{H} = \langle \mathcal{V}, \text{Cost}, \Omega \rangle$ consisting of a hierarchical arena $\mathcal{V} = \langle \mathcal{V}_1, \dots, \mathcal{V}_n \rangle$ equipped with a cost function $\text{Cost} : E \mapsto \{0, 1\}$ for the set of \mathcal{V} -edges, and a coloring mapping $\Omega : N \mapsto C$ for the set of \mathcal{V} -nodes. We can associate to \mathcal{H} an ordinary cost-parity arena (called its *flat expansion*) by recursively substituting each box by a copy of the modular sub-arena it refers to. Since different boxes can refer to the same sub-arena, nodes may appear in different contexts. In general, a state of the flat expansion is a vector whose last component is a node, and the remaining components are boxes that specify the context. Formally, for each modular sub-arena \mathcal{V}_i , we inductively define its flat expansion as the *partial* Cost-parity arena $\mathcal{H}_i^F = \langle \mathcal{A}_i, \text{Cost}_i, \Omega_i, \text{Exit}_i \rangle$, with $\mathcal{A}_i = \langle S_i, S_i^0, S_i^1, R_i, in_i \rangle$, defined as follows:

- The set of states S_i is inductively defined as follows: (i) if u is a node in \mathcal{V}_i , then $u \in S_i$, and (ii) if b is a box of \mathcal{V}_i and $s \in S_{Y_i(b)}$, then $(b, s) \in S_i$.
- S_i^0 (resp., S_i^1) is the set of states in S_i whose node-component belongs to Player 0 (resp., Player 1), and the coloring function Ω_i assigns to each state s of \mathcal{A}_i , the color $\Omega(u)$ of the node-component u of s .
- The transition relation R_i and the cost function Cost_i are inductively defined as follows.
 - If $(u, v) \in E_i$ and the target v is a node, then $(u, v) \in R_i$ and $\text{Cost}_i(u, v) = \text{Cost}(u, v)$. If $(u, b) \in E_i$ and the target b is a box, then $(u, (b, in_{Y_i(b)})) \in R_i$ and $\text{Cost}_i(u, (b, in_{Y_i(b)})) = \text{Cost}(u, b)$.
 - If b is a box of \mathcal{V}_i and $(s, s') \in R_{Y_i(b)}$, then $((b, s), (b, s')) \in R_i$ and $\text{Cost}_i((b, s), (b, s')) = \text{Cost}_{Y_i(b)}(s, s')$.

Note that since $\text{Exit}_1 = \emptyset$, \mathcal{H}_1^F is an ordinary Cost-parity arena (i.e., it is not partial), called the flat expansion of \mathcal{H} . Moreover, observe that each state of \mathcal{H}_1^F is a vector of length at most the nesting depth d of \mathcal{V} , and that the number of states in \mathcal{H}_1^F can be exponential in d . Solving the game on the HCPA \mathcal{H} consists in checking whether the initial state in_1 of the cost-parity arena \mathcal{H}_1^F is winning for Player 0.

3 Solving Hierarchical Cost-Parity Games

The naive approach for solving games on HCPA \mathcal{H} consisting in applying Theorem 1 on the flat expansion of \mathcal{H} would lead to an exponential space procedure. In this section, we show that solving hierarchical cost-parity games is PSPACE-complete. Our approach is based on the notion of *summary function* for a strategy σ of Player 0 in a partial cost-parity arena, which records in a finite and efficient way the overall behavior of all the exit plays of σ with respect to requests and responses. The proposed algorithm for solving the game on the given HCPA \mathcal{H} then solves a sequence of partial cost-parity games, obtained by replacing each box b referring to a sub-arena \mathcal{V}_i with simple partial-cost parity arenas (*summary-gadget arenas*) having the same interface as the flat expansion \mathcal{H}_i^F of \mathcal{V}_i and depending only on the set of colors and exit states. These gadgets represent the behavior of Player 0 as a choice among the possible summary functions associated with the non-losing memoryless strategies in \mathcal{H}_i^F , and also take into account the possibility that the game will stay forever in the sub-arena \mathcal{V}_i for the given context b . The rest of this section is organized as follows: in Subsection 3.1, we introduce the notions of summary and summary-gadget arena, and in Subsection 3.2 we show how to check that a summary is associated with non-losing memoryless strategies. Finally, in Subsection 3.3, we illustrate the proposed algorithm for solving HCPA games.

3.1 Summaries in partial cost-parity games

In this section, for a given partial cost-parity arena \mathcal{G} , we show how to define a finite abstraction of the set of non-losing strategies (of Player 0). Such an abstraction is based on the notion of summary for a strategy σ of Player 0, which is a mapping assigning to each exit state s a value ranging over a finite set (depending only on the set of colors). Such a value summarizes the overall behavior of all the s -exit plays of σ with respect to requests and responses by finitely abstracting the set of associated costs and delays. Then, we associate to each summary \mathcal{S} of \mathcal{G} a simple partial-cost parity arena $Gad(\mathcal{G}, \mathcal{S})$ – exposing the same interface as \mathcal{G} (the initial state and the set of exit states) – which depends only on the set of colors and exit states, and is independent of the set of ‘internal’ states in \mathcal{G} . The set of summary-gadget arenas $Gad(\mathcal{G}, \mathcal{S})$ such that \mathcal{S} is achieved by some non-losing memoryless strategy is ‘context-equivalent’ to \mathcal{G} , i.e., for each memoryless strategy σ achieving some summary \mathcal{S} , \mathcal{G} can be equivalently replaced with $Gad(\mathcal{G}, \mathcal{S})$ in any hierarchical context where \mathcal{G} is exploited as a sub-arena and Player 0 chooses strategy σ when entering \mathcal{G} .²

Fix a partial cost-parity arena $\mathcal{G} = \langle \mathcal{A}, \text{Cost}, \Omega, \text{Exit} \rangle$ over the set C of colors, where $\mathcal{A} = \langle S, S_0, S_1, R, in \rangle$ and Exit is the designated set of exit states.

In order to describe the relative merit of colors, we define an ordering \succeq_0 over the given set C of colors by letting $c \succeq_0 c'$ when c is better for Player 0 than c' . Formally, $c \succeq_0 c'$ if: either (i) c and c' are even and $c \geq c'$, or (ii) c and c' are odd and $c' \geq c$, or (iii) c' is odd and c is even. Moreover, in order to summarize in a finite way cost measures, we exploit three special symbols, namely, bnd_0 , bnd_1 , and unb to denote bounded behavior with zero-cost, bounded behavior with non-zero cost, and unbounded behavior (cost ∞), respectively. Additionally, we denote by \succeq_b the ordering on $\{bnd_0, bnd_1, unb\}$ defined as: $bnd_0 \succeq_b bnd_1$ and $bnd_1 \succeq_b unb$. Intuitively, $bnd_0 \succeq_b bnd_1$ and $bnd_1 \succeq_b unb$ express that bounded zero-cost is better for Player 0 than non-zero bounded cost, the latter being in turn better than unbounded cost. Define $\tilde{C} = C \setminus \{C_o^{max}\}$ and $\tilde{C}_o = (C_o \setminus \{C_o^{max}\}) \cup \{0\}$.

In order to formalize the notion of summary for a strategy σ of Player 0, we consider various cost measures with respect to the requests and the responses along the exit plays of σ . For this, we extend the cost function Cost to (possibly infinite) sets Π of *finite* paths. Formally, $\text{Cost}(\Pi)$ is the least upper bound over the costs of the paths in Π , i.e., $\text{Cost}(\Pi) = \sup\{\text{Cost}(\nu) \mid \nu \in \Pi\}$ where $\sup \emptyset = 0$. Note that $\text{Cost}(\Pi) \in \mathbb{N} \cup \{\infty\}$. For a *finite* path ν of \mathcal{G} and an even color $c_e \in C_e$, a c_e -response in ν is a position k of ν such that $\nu(k)$ has color c_e . For such a response k , the cost of response k in ν is the cost of the prefix of ν leading to position k , i.e., $\text{Cost}(\nu[1, k])$. The c_e -response cost of ν , denoted by $\text{ResCost}(\nu, c_e)$, is the cost $\text{Cost}(\nu[1, k])$ of the prefix of ν up to the minimal c_e -response k in ν for some even color $c'_e \geq c_e$ if such c'_e -responses exist, and it is 0 otherwise. The *maximal even color* of the path ν is the maximal even color visited by ν if ν visits some even color, and it is 0 otherwise (note that a 0-response cannot answer to any request). We exploit the following cost measures for the (possibly infinite) set of exit plays of a given strategy σ of Player 0 leading to a designated exit state.

► **Definition 2** (Cost measures of Player 0 strategies). Let $s \in \text{Exit}$, σ a strategy of Player 0, Π_s the (possibly empty) set of exit plays of σ leading to s , and $c_e \in C_e$ an even color.

- *Cost of σ w.r.t. s* , denoted $\text{Cost}(\sigma, s)$: it is $\text{Cost}(\Pi_s)$.
- *Even c_e -cost of σ w.r.t. s* , denoted $\text{Cost}_e(\sigma, s, c_e)$: it is $\text{Cost}(\Pi_{c_e})$, where Π_{c_e} is the (possibly empty) set of exit plays in Π_s whose maximal even color is at most c_e .

² The formal proof of such a context-equivalence is postponed to Section 3.3.

- c_e -response cost of σ w.r.t. s , denoted $\text{ResCost}(\sigma, s, c_e)$: it is the least upper bound over the c_e -response costs of the exit plays in Π_s , i.e., $\sup\{\text{ResCost}(\nu, c_e) \mid \nu \in \Pi_s\}$.
- Request-cost of σ w.r.t. s , denoted $\text{ReqCost}(\sigma, s)$: it is the least upper bound over the delays associated with the requests along the exit plays in Π_s , i.e., $\sup\{\text{dl}(\nu, k) \mid \nu \in \Pi_s \text{ and } k \text{ is a request in } \nu\}$.

Note that $\text{Cost}_e(\sigma, s, -)$ is monotonic in the third argument, i.e., $\text{Cost}_e(\sigma, s, c'_e) \geq \text{Cost}_e(\sigma, s, c_e)$ for all $c_e, c'_e \in C_e$ such that $c'_e \geq c_e$. We, now, introduce the notion of summary for a strategy σ of Player 0 which records for each exit state s , a value, called *exit value*, ranging over a finite set depending only on the set of colors. This value summarizes the overall behavior of the exit plays of σ leading to s . We distinguish three situations (recall that $C_o^{\max} = \max(C)$ and C_o^{\max} is odd):

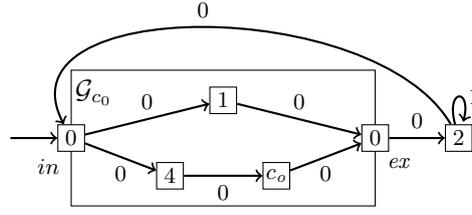
- The best scenario for Player 0 is when there is no exit play of σ leading to s . We represent this situation by exploiting the special symbol \vdash .
- The worst scenario is when the request-cost of σ w.r.t. s is infinite, or there is an s -exit play of σ having a C_o^{\max} -request. We use the color C_o^{\max} to describe this scenario.
- If none of the two previous conditions is fulfilled, then the exit value is a sextuple of elements: (i) the first element summarizes the cost of σ w.r.t. s , (ii) the second element keeps track of the minimal color w.r.t. \preceq_0 over the maximal colors along the s -exit plays of σ , (iii) the third element represents the maximal odd color associated with an unanswered request, and (iv) the last three elements in the tuple summarize the overall response behavior of the s -exit plays of σ .

The formal definition of exit values for a strategy of Player 0 follows.

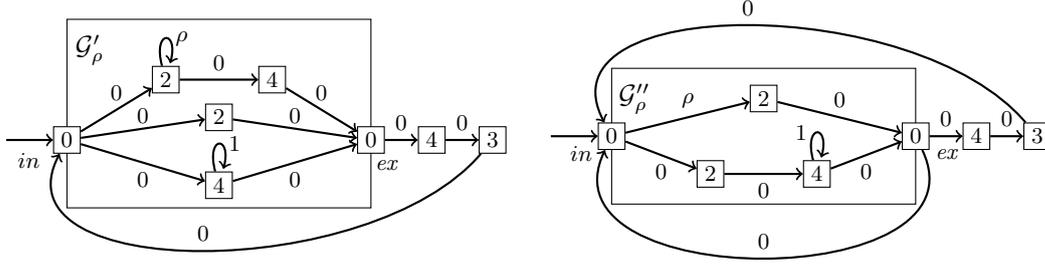
► **Definition 3** (Exit values of Player 0 strategies). Let $s \in \text{Exit}$, σ a strategy of Player 0, and Π_s the set of exit plays of σ leading to s . The *exit value* $\text{value}(\sigma, s)$ of strategy σ w.r.t. s is defined as follows. If $\Pi_s = \emptyset$, then $\text{value}(\sigma, s) = \vdash$. If instead either $\text{ReqCost}(\sigma, s) = \infty$ or there is $\nu \in \Pi_s$ having a C_o^{\max} -request, then $\text{value}(\sigma, s) = C_o^{\max}$. Otherwise, $\text{value}(\sigma, s) = (\text{value}_{\text{Cost}}(\sigma, s), \text{value}_{pr}(\sigma, s), \text{value}_o(\sigma, s), \text{value}_e^L(\sigma, s), \text{value}_e^M(\sigma, s), \text{value}_e^R(\sigma, s)) \in \{\text{bnd}_0, \text{bnd}_1, \text{unb}\} \times \tilde{C} \times \tilde{C}_o \times C_e \times (C_e \cup \{\perp\}) \times C_e$, and the following holds:

- Cost value $\text{value}_{\text{Cost}}(\sigma, s)$: (i) $\text{value}_{\text{Cost}}(\sigma, s) = \text{unb}$ if $\text{Cost}(\Pi_s) = \infty$, (ii) $\text{value}_{\text{Cost}}(\sigma, s) = 0$ if $\text{Cost}(\Pi_s) = 0$, and (iii) $\text{value}_{\text{Cost}}(\sigma, s) = \text{bnd}_1$ otherwise.
- Parity value $\text{value}_{pr}(\sigma, s)$: it is $\min_{\preceq_0}\{c \in C \mid c \text{ is the maximal color of some } \nu \in \Pi_s\}$.
- Odd value $\text{value}_o(\sigma, s)$: it is the *greatest* odd color $c_o \in C_o$ such that for some $\nu \in \Pi_s$, ν has an unanswered c_o -request if such an odd color c_o exists; otherwise, it is 0.
- Even-left value $\text{value}_e^L(\sigma, s)$: it is the *greatest* even color $c_e \in C_e$ such that $\text{ResCost}(\sigma, s, c_e) \neq \infty$ and for each $\nu \in \Pi_s$, the maximal even color in ν is at least c_e , if such an even color c_e exists; otherwise, it is 0.
- Even-middle value $\text{value}_e^M(\sigma, s)$: it is the *smallest* even color c_e such that $\text{Cost}_e(\sigma, s, c_e) \in \mathbb{N} \setminus \{0\}$ if such a color c_e exists, and $\text{value}_e^M(\sigma, s) = \perp$ otherwise (\perp is for ‘undefined’).
- Even-right value $\text{value}_e^R(\sigma, s)$: it is the *greatest* even color $c_e \in C_e$ such that $\text{ResCost}(\sigma, s, c_e) \neq \infty$ and for each $c'_e \in C_e$ with $c'_e < c_e$, $\text{Cost}_e(\sigma, s, c'_e) \neq \infty$, if such an even color c_e exists; otherwise, it is 0.

Note that for parity winning conditions, the parity value $\text{value}_{pr}(\sigma, s)$ suffices for summarizing the s -exit behavior of strategy σ [5]. For cost-parity winning conditions, we also need to keep track of the maximal odd color $\text{value}_o(\sigma, s)$ associated with unanswered requests. Note that $\text{value}_o(\sigma, s) \preceq_0 \text{value}_{pr}(\sigma, s)$, and $\text{value}_o(\sigma, s) \prec_0 \text{value}_{pr}(\sigma, s)$ whenever the maximal unanswered request is associated with s -exit plays whose maximal color is even. As an example, let us consider the sub-arena \mathcal{G}_{c_o} – parametric in the color c_o – in the figure below:



Note that all the states are controlled by Player 1. The instances \mathcal{G}_1 and \mathcal{G}_3 of \mathcal{G}_{c_o} have parity value 1, and odd value 1 and 3, respectively. While by using \mathcal{G}_1 , all the plays starting from state in are winning for Player 0, the same does not hold by using \mathcal{G}_3 since in this case, there are plays where the request 3 is answered in an unbounded way.



For what concerns the even values, the even-left value $value_e^L(\sigma, s)$ represents, intuitively, the maximal even color that the s -exit plays of σ offer for answering – in a bounded way – to previous requests in an arbitrary context. The even-right value $value_e^R(\sigma, s)$, where $value_e^R(\sigma, s) \geq value_e^L(\sigma, s)$, instead represents the maximal even color which may answer to a request preceding an s -exit play ν of σ in a bounded way: if the maximal even color in ν is smaller than $value_e^R(\sigma, s)$, then the overall cost of all s -exit plays of σ whose maximal even color is smaller than $value_e^R(\sigma, s)$ is finite. As an example, let us consider the sub-arena \mathcal{G}'_ρ – parametric in the cost ρ of the self-loop on the state with color 2 – in the left part of the figure above, where all the states are controlled by Player 1. The instances \mathcal{G}'_0 and \mathcal{G}'_1 of \mathcal{G}'_ρ have even-left value 2, and even-right value 4 and 2, respectively. While for \mathcal{G}'_0 , all the plays starting from state in are winning for Player 0, the same does not hold for \mathcal{G}'_1 , since in this case, there are plays where the external request 3 is answered in an unbounded way. Finally, in order to illustrate the importance of the even-middle value, let us consider the sub-arena \mathcal{G}''_ρ in the right part of the figure above, where again all the states are controlled by Player 1. The instances \mathcal{G}''_0 and \mathcal{G}''_1 of \mathcal{G}''_ρ have even-left value 2, right-even value 4, and even-middle value \perp and 2, respectively. While for \mathcal{G}''_0 , all the plays starting from in are winning for Player 0, for \mathcal{G}''_1 , there are plays where the external request 3 is answered in an unbounded way. We make the following observations which easily follow from Definition 3.

► **Proposition 4.** *Let σ be a strategy of Player 0 in \mathcal{G} and $s \in Exit$ such that $value(\sigma, s) = (f, c_{pr}, c_o, c_e^L, c_e^M, c_e^R)$. Then:*

- $c_o \preceq_o c_{pr}$, $c_e^L \leq c_{pr}$, $c_e^L \leq c_e^R$, and $c_e^M \in [c_e^L, c_e^R]$ if $c_e^M \neq \perp$.
- $c_e^M = \perp$ if $f = bnd_0$, and $c_e^M \neq \perp$ if $f = bnd_1$.
- $c_e^R = \max(C_e)$ if $f \neq unb$, and $c_e^R < \max(C_e)$ if $f = unb$ and $c_e^M = c_e^R$.
- $c_e^L = c_{pr}$ if $c_{pr} \in C_e$ and either $f \neq unb$, or $c_e^L < c_e^R$, or $c_e^M = c_e^R$.

► **Definition 5 (Summaries of Player 0 strategies).** The set \mathcal{E}_C of *exit values* for the set C of colors is the finite set $\{\perp, C_o^{max}\} \cup \mathcal{E}'_C$, where \mathcal{E}'_C is the set of tuples $(f, c_{pr}, c_o, c_e^L, c_e^M, c_e^R) \in \{bnd_0, bnd_1, unb\} \times \tilde{C} \times \tilde{C}_o \times C_e \times (C_e \cup \{\perp\}) \times C_e$ satisfying Conditions (1)–(4) in Proposition 4.

A *summary* of \mathcal{G} is a mapping $\mathcal{S} : Exit \mapsto \mathcal{E}_C$ such that for all $s \in Exit$ with $\mathcal{S}(s) = (f, c_{pr}, c_o, c_e^L, c_e^M, c_e^R)$, it holds that $c_e^L \succeq_0 \Omega(in)$, $c_o \preceq_0 \Omega(s)$, and $\Omega(in) \leq c_e^R$. The summary

6:10 Hierarchical Cost-Parity Games

$\mathcal{S}(\sigma)$ of a strategy σ of Player 0 in \mathcal{G} is the summary of \mathcal{G} associating to each $s \in \text{Exit}$, the exit value $\text{value}(\sigma, s)$.

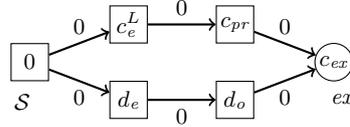
For each summary \mathcal{S} , we now define a partial-cost parity game $\text{Gad}(\mathcal{G}, \mathcal{S})$, exposing the same interface as \mathcal{G} and independent of the set of ‘internal’ states in \mathcal{G} , such that there is a unique strategy $\sigma_{\mathcal{S}}$ of Player 0 in $\text{Gad}(\mathcal{G}, \mathcal{S})$. Moreover, $\sigma_{\mathcal{S}}$ is non-losing and the exit values of $\sigma_{\mathcal{S}}$ correspond to the exit values of any strategy of Player 0 in \mathcal{G} having \mathcal{S} as summary.

► **Definition 6 (Summary-Gadget Arena).** Let \mathcal{S} be a summary of \mathcal{G} . Given $ex \in \text{Exit}$, we first define the sub-gadget $\text{Gad}(\mathcal{G}, \mathcal{S}, ex)$ of \mathcal{G} for summary \mathcal{S} and ex , which is the partial cost-parity game with set of states $S_{ex} \cup \{\mathcal{S}, ex\}$ and set of edges R_{ex} , where:

- All the states in $S_{ex} \cup \{\mathcal{S}\}$ are controlled by Player 1, \mathcal{S} has color 0 and is the initial state, ex is the unique exit state, and the color and the player of state ex is as in \mathcal{G} .

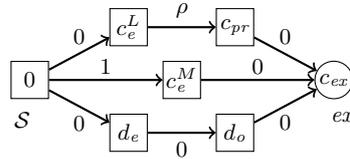
Moreover, if $\mathcal{S}(ex) = \perp$, then $S_{ex} = \emptyset$, and $R_{ex} = \emptyset$. On the opposite side, if $\mathcal{S}(ex) = C_o^{max}$, then S_{ex} consists of a unique state s having color C_o^{max} , and R_{ex} consists of two edges, one from state \mathcal{S} to state s with cost 0, and the other one from s to ex with cost 0 as well. Otherwise, let $\mathcal{S}(ex) = (f, c_{pr}, c_o, c_e^L, c_e^M, c_e^R)$. Then, we distinguish six cases, where (i) c_{ex} is the color of ex , and (ii) $d_o = c_{pr}$ and $d_e = c_e^L$ if $c_o \in \{0, c_{pr}\}$, and $d_o = c_o$ and $d_e = \max(\{c_e, c_o + 1\})$ otherwise, where $c_e = c_e^L$ if $c_{pr} \in C_o$, and $c_e = c_{pr}$ otherwise. In the figures illustrating the construction, we assume that ex is controlled by Player 0.

Case $f = bnd_0$



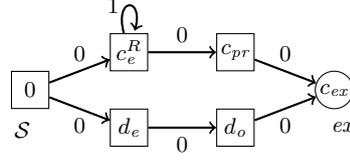
In this case, we have $c_e^M = \perp$ and $c_e^R = \max(C_e)$. The sub-gadget $\text{Gad}(\mathcal{G}, \mathcal{S}, ex)$ for this case is a DAG and is illustrated on the left. Note that the cost of any path from state \mathcal{S} to the exit state ex is 0.

Case $f = bnd_1$



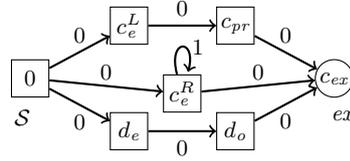
In this case, we have that $c_e^M \in C_e$, $c_e^M \in [c_e^L, c_e^R]$, and $C_e^R = \max(C_e)$. The associated sub-gadget is a DAG and it is illustrated on the right, where $\rho = 0$ if $c_e^L < c_e^M$, and $\rho = 1$ otherwise. Note that the overall cost of all paths from state \mathcal{S} to the exit state ex is 1. Moreover, according to the definition of even-middle value, c_e^M represents the smallest even color c_e such that the cost of all exit plays leading to ex and having maximal even color c_e is finite and non-null. Additionally, if $c_e^L < c_e^M$, according to the definition of even-left value, there are exit plays leading to ex whose maximal even color is c_e^L , and the overall cost of such exit plays is 0.

Case $f = unb$, $c_e^M = \perp$, and $c_e^L = c_e^R$



The sub-gadget $Gad(\mathcal{G}, S, ex)$ for this case is illustrated on the left. When $f = unb$, the overall cost of all exit plays leading to ex is infinite. This is implemented by a self-loop with cost 1 on the state having color c_e^R . Note that for a strategy σ of Player 0 with $value_{Cost}(\sigma, ex) = unb$ and $value_e^R(\sigma, ex) = c_e^R$, the overall cost of all ex -exit plays having maximal even color at most c_e^R maybe finite. However, in this case, $c_e^R < \max(C_e)$ and $ResCost(\sigma, ex, c_e^R + 2) = \infty$. Thus, the self-loop with cost 1 in the sub-gadget above takes into account also these possible scenarios.

Case $f = unb$, $c_e^M = \perp$, and $c_e^L < c_e^R$



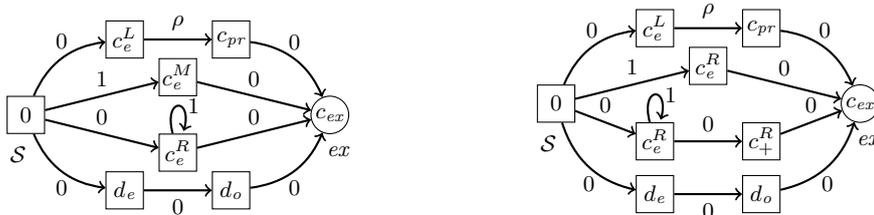
This case is similar to the previous one. The unique difference is that now $c_e^L < c_e^R$. Thus, the associated sub-gadget – illustrated on the right – summarizes strategies σ of Player 0 for which, in particular, $value_e^L(\sigma, ex) = c_e^L$ and there are exit plays leading to ex whose maximal even color is c_e^L , and the overall cost of such exit plays is 0.

Case $f = unb$, $c_e^M \in C_e$, and $c_e^M < c_e^R$

This case is similar to the previous one, but now $c_e^M \in C_e$, hence, $c_e^M \in [c_e^L, c_e^R]$. The associated sub-gadget is illustrated in the left part of the figure below, where $\rho = 0$ if $c_e^L < c_e^M$, and $\rho = 1$ otherwise.

Case $f = unb$, $c_e^M \in C_e$, and $c_e^M = c_e^R$

In this case, we have that $c_e^M \in [c_e^L, c_e^R]$ and $c_e^R < \max(C_e)$. The associated sub-gadget is illustrated on the right of the figure below, where $c_+^R = c_e^R + 2$, $\rho = 0$ if $c_e^L < c_e^M$, and $\rho = 1$ otherwise. In this case there is an even color, namely c_+^R , whose response-cost with respect to ex is infinite. This is consistent with the fact that for all strategies σ of Player 0 such that $value_{Cost}(\sigma, ex) = unb$, $value_e^R(\sigma, ex) = c_e^R$, and $Cost_e(\sigma, ex, c_e^R) \neq \infty$, we have that $ResCost_e(\sigma, ex, c_e^R + 2) = \infty$.



We now define the gadget arena $Gad(\mathcal{G}, S)$ for the given summary \mathcal{S} , which is intuitively obtained by merging the sub-gadgets $Gad(\mathcal{G}, S, ex)$ for the various exit states $ex \in Exit$ and

by adding the state in . Formally, assuming that $S_{ex} \cap S_{ex'} = \emptyset$ (i.e., sub-gadgets associated with distinct exit states share only state \mathcal{S}), $Gad(\mathcal{G}, \mathcal{S})$ has the same interface as \mathcal{G} and satisfies the following: the set of states of $Gad(\mathcal{G}, \mathcal{S})$ is $\{in, \mathcal{S}\} \cup Exit \cup \bigcup_{ex \in Exit} S_{ex}$ and the set of transitions is $\{(in, \mathcal{S})\} \cup \bigcup_{ex \in Exit} R_{ex}$, where transition (in, \mathcal{S}) has cost 0.

► **Remark.** Note that in a summary-gadget arena $Gad(\mathcal{G}, \mathcal{S})$, every state which is not in $\{in\} \cup Exit$ is controlled by Player 1. In particular, there is exactly one strategy of Player 0, and such a strategy is non-losing.

By construction, we easily obtain the following result.

► **Proposition 7.** *Let $\mathcal{G} = \langle \mathcal{A}, Cost, \Omega, Exit \rangle$ be a partial-cost parity arena, σ a strategy of Player 0, σ_S the unique strategy of Player 0 in $Gad(\mathcal{G}, \mathcal{S}(\sigma))$, and $s \in Exit$. Then, $value(\sigma, s) = value(\sigma_S, s)$. Moreover, if $value(\sigma, s) \neq C_o^{max}$, the following holds:*

- *Let ν be an s -exit play of σ_S with maximal even color c_e . Then, either (i) $Cost_e(\sigma, s, c_e) \geq Cost_e(\sigma_S, s, c_e)$, and there is an s -exit play ν' of σ whose maximal even color is at most c_e , or (ii) $Cost_e(\sigma_S, s, c_e) = \infty$, $c_e < \max(C_e)$, and $ResCost(\sigma, s, c_e + 2) = \infty$.*
- *For each $c_e \in C_e$, $ResCost(\sigma_S, s, c_e) = \infty$ entails that $ResCost(\sigma, s, c_e) = \infty$.*

Not all the summaries of \mathcal{G} are associated with non-losing strategies (of Player 0). On the other hand, checking whether a summary is associated with a non-losing strategy is not an easy task since we have to check the fulfillment of unboundedness conditions. However, we can get around the problem by exploiting monotonicity properties of the cost-parity winning conditions. We define a reflexive and transitive relation \sqsupseteq over the set of summaries. Intuitively, $\mathcal{S} \sqsupseteq \mathcal{S}'$ when \mathcal{S} is not worse than \mathcal{S}' for Player 0. A summary \mathcal{S} is then *relevant* if $\mathcal{S}(\sigma) \sqsupseteq \mathcal{S}$ for some non-losing *memoryless* strategy σ . As we will see in Section 3.2, checking whether a summary is relevant can be done in polynomial space.

► **Definition 8** (Relevant summaries). Let \sqsupseteq be a binary relation over \mathcal{E}_C defined as follows:

- $\vdash \sqsupseteq ev$ for all $ev \in \mathcal{E}_C$;
- $ev \sqsupseteq C_o^{max}$ for all $ev \in \mathcal{E}_C$;
- $(f, c_{pr}, c_o, c_e^L, c_e^M, c_e^R) \sqsupseteq (\tilde{f}, \tilde{c}_{pr}, \tilde{c}_o, \tilde{c}_e^L, \tilde{c}_e^M, \tilde{c}_e^R)$ if $f \succeq_b \tilde{f}$, $c_{pr} \succeq_0 \tilde{c}_{pr}$, $c_o \succeq_0 \tilde{c}_o$, $c_e^L \succeq_0 \tilde{c}_e^L$, $c_e^R \succeq_0 \tilde{c}_e^R$, and the following holds:
 - if $c_e^M \neq \perp$, then either $\tilde{c}_e^M \neq \perp$ and $c_e^M \geq \tilde{c}_e^M$, or $\tilde{c}_e^M = \perp$ and $c_e^M \geq \tilde{c}_e^R$.

Given two summaries \mathcal{S} and \mathcal{S}' of \mathcal{G} , we say that \mathcal{S} is *not worse than \mathcal{S}' for Player 0*, written $\mathcal{S} \sqsupseteq \mathcal{S}'$, if $\mathcal{S}(s) \sqsupseteq \mathcal{S}'(s)$ for all $s \in Exit$. A summary \mathcal{S} of \mathcal{G} is *relevant* iff there is a memoryless non-losing strategy σ in \mathcal{G} such that $\mathcal{S}(\sigma) \sqsupseteq \mathcal{S}$.

► **Remark.** The binary relation \sqsupseteq over the set of summaries is reflexive and transitive.

Note that if \mathcal{G} has no exits, then the unique summary is the empty set, and such a summary is relevant iff there is a memoryless winning strategy of Player 0 from in . By construction, we easily obtain the following result, which represents the converse of Proposition 7.

► **Proposition 9.** *Let $\mathcal{G} = \langle \mathcal{A}, Cost, \Omega, Exit \rangle$ be a partial-cost parity arena, \mathcal{S} a summary of \mathcal{G} , σ a strategy of Player 0 such that $\mathcal{S}(\sigma) \sqsupseteq \mathcal{S}$, σ_S the unique strategy of Player 0 in $Gad(\mathcal{G}, \mathcal{S})$, and $s \in Exit$. Then $value(\sigma, s) \sqsupseteq value(\sigma_S, s)$. Moreover, if $\mathcal{S}(s) \neq C_o^{max}$, the following holds:*

- *Let ν be an s -exit play of σ , c_e the maximal even color of ν , and $Cost_e(\sigma, s, c_e) = m \in \mathbb{N} \cup \{\infty\}$. Then, either (i) $Cost_e(\sigma_S, s, c_e) = m'$ where $m' > 0$ if $m > 0$, and $m' = \infty$ if $m = \infty$, and there is a s -exit play ν' of σ_S whose maximal even color is at most c_e , or (ii) $Cost_e(\sigma, s, c_e) = \infty$, $c_e < \max(C_e)$, and $ResCost(\sigma_S, s, c_e + 2) = \infty$.*

- For each $c_e \in C_e$, if $\text{ResCost}(\sigma, s, c_e) = \infty$, one of the following holds:
 - either $\text{ResCost}(\sigma_S, s, c'_e) = \infty$ for some even color $c'_e \leq c_e$,
 - or there is an even color $c'_e \leq \text{value}_e^R(\sigma, s) < c_e$ such that $\text{Cost}_e(\sigma_S, s, c'_e) = \infty$.

Note that the set of relevant summaries in \mathcal{G} is empty iff there does not exist a memoryless non-losing strategy in \mathcal{G} . By Theorem 1, checking this condition can be done in polynomial space. In this case, we associate with \mathcal{G} a simple partial cost-parity arena (bad gadget), where Player 0 always loses.

► **Definition 10** (Bad-Gadget Arena). The *bad-gadget arena* $\text{BadGad}(\mathcal{G})$ of \mathcal{G} is the partial cost-parity game having the same interface as \mathcal{G} and defined as follows: $\text{BadGad}(\mathcal{G})$ has a unique ‘internal’ state $s \notin \{in\} \cup \text{Exit}$, which has color 0 and is controlled by Player 0, and a unique transition, namely (in, s) , which has cost 0.

3.2 Checking relevance of summaries

We reduce the problem of checking summary relevance in partial cost-parity arenas to verifying the existence of *memoryless* strategies in cost-parity arenas under a simple imperfect-information setting. Formally, an *observation-based cost-parity arena* (OCPA) is a cost-parity arena $\mathcal{G} = \langle \mathcal{A}, \text{Cost}, \Omega, \text{Obs} \rangle$ equipped with an observability equivalence relation $\text{Obs} \subseteq S \times S$ over the set of states. An *observation-based memoryless* strategy of Player 0 is a memoryless strategy σ of Player 0 such that, for all non-terminal states s and s' controlled by Player 0, $(s, s') \in \text{Obs} \Rightarrow (\sigma(s), \sigma(s')) \in \text{Obs}$. The following easily follows.

► **Theorem 11.** *Let $\mathcal{G} = \langle \mathcal{A}, \text{Cost}, \Omega, \text{Obs} \rangle$ be an OCPA. Checking the existence of a winning observation-based memoryless strategy of Player 0 from the initial state can be done in polynomial space.*

► **Theorem 12** (Checking summary relevance). *Let $\mathcal{G} = \langle \mathcal{A}, \text{Cost}, \Omega, \text{Exit} \rangle$ be a partial cost-parity arena over C with $\mathcal{A} = \langle S, S_0, S_1, R, in \rangle$ and \mathcal{S} a summary of \mathcal{G} . Then, one can check in polynomial space whether \mathcal{S} is relevant.*

Proof. We build in polynomial time an OCPA \mathcal{G}_S such that, there is a winning observation-based memoryless strategy of Player 0 in \mathcal{G}_S from the initial state iff \mathcal{S} is relevant in \mathcal{G} . We first construct a partial OCPA \mathcal{G}' obtained from \mathcal{G} by extending every state of \mathcal{G} with additional information which keeps tracks of the maximal even color and the maximal *unanswered* odd color visited in the current play-prefix from in and a flag indicating whether such a prefix has cost zero. Formally, $\mathcal{G}' = \langle \mathcal{A}', \text{Cost}', \Omega', \text{Exit}', \text{Obs} \rangle$ where $\mathcal{A}' = \langle S', S'_0, S'_1, R', in' \rangle$ and:

- $S' = S \times C_e \times \tilde{C}_o \times \{0, 1\}$, $\text{Exit}' = \text{Exit} \times C_e \times \tilde{C}_o \times \{0, 1\}$, $in' = (in, 0, 0, 0)$, $\Omega'((s, c_e, c_o, d)) = \Omega(s)$, and $((s, c_e, c_o, d), (s', c'_e, c'_o, d')) \in \text{Obs}$ iff $s = s'$. Moreover, the player of each state (s, c_e, c_o, d) is the player of s in \mathcal{G} if $s \notin \text{Exit}$, Player 0 if $s \in \text{Exit}$ and $\mathcal{S}(s) = \vdash$, and Player 1 otherwise.
- $((s, c_e, c_o, d), (s', c'_e, c'_o, d')) \in E'$ iff (i) $(s, s') \in E$, (ii) $c'_e = \max_{\geq 0}(\{c_e, \Omega(s)\})$, (iii) $c'_o = 0$ if $\Omega(s') \in C_e$ and $\Omega(s') \geq c_o$, and $c_o = \min_{\geq 0}(\{c_o, \Omega(s)\})$ otherwise, and (iv) $d' = 0$ if $d = 0$ and $\text{Cost}(s, s') = 0$, and $d' = 1$ otherwise;
- $\text{Cost}'((s, c_e, c_o, d), (s', c'_e, c'_o, d')) = \text{Cost}(s, s')$.

Note that by construction, there is a bijection, denoted by Obs , between the memoryless strategies σ of Player 0 in \mathcal{G} , and the observation-based memoryless strategies of Player 0 in \mathcal{G}' . Formally, for each non-terminal state (s, c_e, c_o, d) of \mathcal{G}' controlled by Player 0, $\text{Obs}(\sigma)((s, c_e, c_o, d))$ is the unique successor of (s, c_e, c_o, d) having as S-component $\sigma(s)$.

For each $ex \in \text{Exit}$, let Exit'_{ex} be the set of exit states of \mathcal{G}' having ex as S-component. The game \mathcal{G}_S is obtained from \mathcal{G}' by adding for each exit state $ex \in \text{Exit}$ such that $\mathcal{S}(ex) \notin \{\vdash, C_o^{\max}\}$, a gadget (subgraph) consisting of states controlled by Player 1 that connects the exit states of \mathcal{G}' in Exit'_{ex} with the initial state $in' = (in, 0, 0, 0)$, and an additional terminal state \square which is controlled by Player 0. If $\mathcal{S}(ex) = \vdash$, then for every strategy σ of Player 0 in \mathcal{G} , $\mathcal{S}(\sigma)(ex) = \vdash$ if $\mathcal{S}(\sigma) \sqsupseteq \mathcal{S}$, and our choice (states in Exit'_{ex} are controlled by Player 0) allows to capture only the non-losing strategies σ of \mathcal{G} for which there is no exit play leading to ex . On the other hand, if $\mathcal{S}(ex) = C_o^{\max}$, then for each strategy σ of Player 0 in \mathcal{G} , $\mathcal{S}(\sigma)(ex) \sqsupseteq C_o^{\max}$, and accordingly, states in Exit'_{ex} are controlled by Player 1.

Now, we describe the construction of the gadget for ex when $\mathcal{S}(ex) \notin \{\vdash, C_o^{\max}\}$, i.e., $\mathcal{S}(ex)$ is of the form $(f, c_{pr}, c_o, c_e^L, c_e^M, c_e^R) \in \{bnd_0, bnd_1, unb\} \times \tilde{C} \times \tilde{C}_o \times C_e \times (C_e \cup \{\perp\}) \times C_e$. Due to space limitations, here we focus only on the case where $f = unb$ and $c_o > c_e^R$. Note that for each strategy σ of Player 0, it holds that $\text{Cost}(\sigma, ex) \succeq_b unb$. The gadget for this case is obtained by adding 2 new states *controlled by Player 1*, namely ex_e and ex_o^R , a new terminal state \square controlled by Player 0, and new transitions. State \square has color 0, state ex_e has the even color $c_o + 1$, and state ex_o^R has color 0 if $c_e^R = 0$, and the odd color $c_e^R - 1$ otherwise. The new transitions have cost 0 and are as follows:

- for each $s = (ex, c'_e, c'_o, d) \in \text{Exit}'_{ex}$ such that one of the following bad conditions is satisfied, we add the transition (s, \square) .
 - *Bad conditions:* either (i) $\max(\{c'_o, c'_e\}) \prec_o c_{pr}$, or (ii) $c'_e < c_e^L$, or (iii) $d = 1$, $c_e^M = \perp$, and $c'_e < c_e^R$, or (iv) $d = 1$, $c_e^M \neq \perp$, and $c'_e < c_e^M$.
- the transitions (ex_e, ex_o^R) and (ex_o^R, in') , and for each $s \in \text{Exit}'_{ex}$, the transition (s, ex_e) .

The transitions having as target state \square are exploited to capture the strategies σ of Player 0 in \mathcal{G} satisfying the following: (i) $\text{value}_{pr}(\sigma, ex) \succeq_o c_{pr}$, (ii) in each exit play ν of σ leading to ex , the maximal even color of ν is at least c_e^L and (iii) if $\text{value}_e^M(\sigma, ex) \neq \perp$, then either $c_e^M = \perp$ and $\text{value}_e^M(\sigma, ex) \geq c_e^R$, or $c_e^M \neq \perp$ and $\text{value}_e^M(\sigma, ex) \geq c_e^M$.

Moreover, given a memoryless strategy σ of Player 0 in \mathcal{G} , the chains of transitions (s, ex_e) , (ex_e, ex_o^R) and (ex_o^R, in') entering the initial state in' , where $s \in \text{Exit}'_{ex}$, are responsible of cycles consistent with $\text{Obs}(\sigma)$ of the form $\nu \cdot ex_e \cdot ex_o^R \cdot in'$, where ν is an arbitrary exit play of $\text{Obs}(\sigma)$ leading to some exit state $s \in \text{Exit}'_{ex}$. By concatenating these cycles, one obtains infinite plays consistent with $\text{Obs}(\sigma)$ which are winning for Player 0 iff $\text{value}(\sigma, ex) \neq C_o^{\max}$ (the request cost of σ w.r.t. 0 is finite), $\text{value}_o(\sigma, ex) \succeq_0 c_o$, and $\text{value}_e^R(\sigma, ex) \geq c_e^R$.

By construction, for each memoryless strategy σ of Player 0 in \mathcal{G} , σ is non-losing and $\mathcal{S}(\sigma) \sqsupseteq \mathcal{S}$ iff $\text{Obs}(\sigma)$ is winning for Player 0 from state in' . Thus, since Obs is a bijection between the memoryless strategies of Player 0 in \mathcal{G} and the observation-based memoryless strategies of Player 0 in \mathcal{G}_S , by Theorem 11, Theorem 12 follows. ◀

3.3 Algorithm for solving games on HCPA

In this section, by exploiting the summary-gadget arena construction of Section 3.1, we derive a polynomial space algorithm for solving hierarchical cost-parity games. In particular, we describe an NPSpace procedure which solves the considered problem (recall that by Savitch's theorem, PSPACE = NPSpace). The outline of the nondeterministic procedure, called Algorithm 1, is given in Fig. 1.

Given an HCPA $\mathcal{H} = \langle \mathcal{V}, \text{Cost}, \Omega \rangle$ with $\mathcal{V} = \langle \mathcal{V}_1, \dots, \mathcal{V}_n \rangle$, Algorithm 1 proceeds in phases corresponding to the iterations of the repeat loop. In each phase, the modular sub-arenas of \mathcal{H} are processed in increasing order w.r.t. the hierarchical level, starting with the lowest level sub-arena \mathcal{V}_n which has no boxes, and, therefore, corresponds to its flat expansion \mathcal{H}_n^F . At

Algorithm 1

Input: $\mathcal{H} = \langle \langle \mathcal{V}_1, \dots, \mathcal{V}_n \rangle, \text{Cost}, \Omega \rangle$

repeat

for $i = n$ **downto** 1 **do**

guess $\mathcal{G}_{app,i} \in \text{AppSimplify}(\mathcal{V}_i, g_{app,i});$

$g_{app,i}(b) = \mathcal{G}_{app,Y_i(b)}$ for all $b \in B_i;$

if Player 0 wins in $\mathcal{G}_{app,1}$ **then accepts**

■ **Figure 1** NPSPACE procedure.

step $n \geq i \geq 1$ of the current phase, the algorithm nondeterministically chooses a partial cost-parity arena $\mathcal{G}_{app,i}$ from a finite set of partial cost-parity arenas obtained by applying the operation *AppSimplify* (approximated simplification) to the modular sub-arena \mathcal{V}_i and the *substitution* mapping $g_{app,i}$. A substitution for a modular sub-arena \mathcal{V}_i is a mapping associating to each box b of \mathcal{V}_i with $Y_i(b) = k$, a partial cost-parity arena having the same interface as the flat expansion \mathcal{H}_k^F of \mathcal{V}_k (hence, $g(b)$ has initial state in_k and set of exit states $Exit_k$). In our case, the substitution $g_{app,i}$ considered by the algorithm at iteration $n \geq i \geq 1$ maps each box b of \mathcal{V}_i with the the guessed approximation $\mathcal{G}_{app,Y_i(b)}$ in the previous iteration $Y_i(b)$ of the current phase (recall that $Y_i(b) > i$). The essence of the operation *AppSimplify* is to replace each box b of \mathcal{V}_i with a copy of the summary-gadget arena of $\mathcal{G}_{app,Y_i(b)}$ associated with some relevant summary of $\mathcal{G}_{app,Y_i(b)}$. Note that at iteration n , $g_{app,n}$ is empty and $\mathcal{G}_{app,n}$ coincides with \mathcal{H}_n^F . If Player 0 wins in $\mathcal{G}_{app,1}$ (recall that the top-level arena \mathcal{V}_1 has no exit, hence, $\mathcal{G}_{app,1}$ has no exit as well), then the algorithm accepts the input \mathcal{H} . Otherwise, a new phase is started. We now formally define the approximated simplification operation.

► **Definition 13** (Approximated simplification). Let $\mathcal{H} = \langle \mathcal{V}, \text{Cost}, \Omega \rangle$ be an *HCPA* with $\mathcal{V} = \langle \mathcal{V}_1, \dots, \mathcal{V}_n \rangle$, $i \in [1, n]$, and g be a substitution for \mathcal{V}_i . For a box b of \mathcal{V}_i and a relevant summary \mathcal{S} of $g(b)$, we denote by $Gad_b(g(b), \mathcal{S})$ the copy of the summary-gadget arena $Gad(g(b), \mathcal{S})$ associated with $g(b)$ and \mathcal{S} obtained by replacing each state s in $Gad(g(b), \mathcal{S})$ with the copy (b, s) . The b -copy $BadGad_b(g(b))$ of the bad-gadget arena $BadGad(g(b))$ for $g(b)$ is defined in a similar way. Note that the copies of the states in $\{in_k\} \cup Exit_k$, where $k = Y_i(b)$, are states in the flat expansion \mathcal{H}_i^F of \mathcal{V}_i .

The *simplification* $\text{Simplify}(\mathcal{V}_i, g, b)$ of \mathcal{V}_i w.r.t. the substitution g and the box b (resp., the *simplification* $\text{Simplify}(\mathcal{V}_i, g, b, \mathcal{S})$ of \mathcal{V}_i w.r.t. the substitution g , the box b , and a relevant summary \mathcal{S} of $g(b)$) is the partial cost-parity arena obtained from \mathcal{H}_i^F as follows:

- all the states in \mathcal{H}_i^F of the form (b, s) which are not in $BadGad_b(g(b))$ (resp, $Gad_b(g(b), \mathcal{S})$) are removed together with the associated transitions, and all the states in $BadGad_b(g(b))$ (resp, $Gad_b(g(b), \mathcal{S})$) are added together with the associated transitions.

An *approximated simplification* of \mathcal{V}_i w.r.t. g is a partial cost-parity arena obtained by applying for each box b of \mathcal{V}_i , the simplification operation w.r.t. g and b if the set of relevant summaries of $g(b)$ is empty, and the simplification operation w.r.t. g , b , and some relevant summary \mathcal{S}_b of $g(b)$ otherwise. We denote by $\text{AppSimplify}(\mathcal{V}_i, g)$ the set of approximated simplifications of \mathcal{V}_i w.r.t. g .

Note that the arenas in $\text{AppSimplify}(\mathcal{V}_i, g)$ can be constructed directly from \mathcal{V}_i without constructing the flat expansion \mathcal{H}_i^F . By Propositions 7 and 9, we deduce that the *AppSimplify* operation preserves the set of relevant summaries. In particular, the following holds, where for a partial cost-parity arena \mathcal{G} , $RS(\mathcal{G})$ is the set of relevant summaries in \mathcal{G} (we extend the notation RS to sets of partial cost-parity arenas).

► **Lemma 14.** *Let $\mathcal{H} = \langle \mathcal{V}, \text{Cost}, \Omega \rangle$ with $\mathcal{V} = \langle \mathcal{V}_1, \dots, \mathcal{V}_n \rangle$ be an HCPA, $i \in [1, n]$, and g a substitution for \mathcal{V}_i . Assume that for each box b of \mathcal{V}_i , $RS(\mathcal{H}_{\mathcal{V}_i(b)}^F) = RS(g(b))$. Then, $RS(\mathcal{H}_i^F) = RS(\text{AppSimplify}(\mathcal{V}_i, g))$.*

By Lemma 14, we deduce the main result of this paper.

► **Theorem 15.** *Solving hierarchical cost-parity games is PSPACE-complete.*

4 Conclusion

Cost-parity games represent a powerful machinery for the verification of temporal requirements that are bounded in time. As in many settings, the representation of systems by means of cost-parity games is affected by an exponential blow-up in the size of the resulting game. To overcome this, many techniques exploiting system regularities have been successfully applied. Among them, hierarchical systems deserve a special mention. In this paper, we have introduced and investigated the problem of solving cost-parity games over hierarchical FSMs, showing that the problem is PSPACE-complete, thus not harder than solving parity games over hierarchical models. As future work, we aim to adapt the proposed approach to all the other winning bounded conditions introduced in [24]. Moreover, it would be interesting to investigate cost-parity conditions over concurrent game structures, the last one being a suitable formalism for modelling strategic environments where there is simultaneous interaction between multiple players. Other relevant research directions include the study of cost-parity games in the imperfect information setting as well as for infinite-state systems.

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, M. Benedikt, K. Etessami, P. Godefroid, T. W. Reps, and M. Yannakakis. Analysis of recursive state machines. *TOPLAS*, 27(4):786–818, 2005.
- 3 R. Alur, S. Kannan, and M. Yannakakis. Communicating hierarchical state machines. In *Automata, Languages and Programming, 26th International Colloquium, ICALP'99, Prague, Czech Republic, July 11-15, 1999, Proceedings*, pages 169–178, 1999.
- 4 R. Alur and M. Yannakakis. Model checking of hierarchical state machines. *ACM Trans. Program. Lang. Syst.*, 23(3):273–303, 2001.
- 5 B. Aminof, O. Kupferman, and A. Murano. Improved Model Checking of Hierarchical Systems. *Inf. Comput.*, 210:68–86, 2012. doi:10.1016/j.ic.2011.10.008.
- 6 B. Aminof, F. Mogavero, and A. Murano. Synthesis of Hierarchical Systems. In *FACS'11*, LNCS 7253, pages 42–60. Springer, 2011.
- 7 B. Aminof, F. Mogavero, and A. Murano. Synthesis of Hierarchical Systems. *SCP*, 83:56–79, 2014.
- 8 K. Chatterjee and L. Doyen. Energy Parity Games. *Theor. Comput. Sci.*, 458:49–60, 2012.
- 9 K. Chatterjee, T. A. Henzinger, and M. Jurdzinski. Mean-Payoff Parity Games. In *LICS'05*, pages 178–187, 2005.
- 10 K. Chatterjee, T. A. Henzinger, and F. Horn. Finitary winning in ω -regular games. *ACM Trans. Comput. Logic*, 11(1), 2009.
- 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. Springer, 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*, pages 368–377, 1991.

- 14 E. A. Emerson and C. S. Jutla. The Complexity of Tree Automata and Logics of Programs. *SJM*, 29(1):132–158, 1999.
- 15 N. Fijalkow and M. Zimmermann. Parity and Streett Games with Costs. *Logical Methods in Computer Science*, 10(2), 2014.
- 16 S. Göller and M. Lohrey. Fixpoint logics on hierarchical structures. In *FSTTCS'05*, LNCS 3821, pages 483–494. Springer, 2005.
- 17 M. Jurdzinski, M. Paterson, and U. Zwick. A Deterministic Subexponential Algorithm for Solving Parity Games. *SIAM J. Comput.*, 38(4):1519–1532, 2008.
- 18 D. Kozen. Results on the Propositional muCalculus. *TCS*, 27(3):333–354, 1983.
- 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. Y. Vardi, and P. Wolper. An Automata Theoretic Approach to Branching-Time Model Checking. *JACM*, 47(2):312–360, 2000.
- 21 O. Kupferman, M. Y. Vardi, and P. Wolper. Module Checking. *IC*, 164(2):322–344, 2001.
- 22 V. Malvone, A. Murano, and L. Sorrentino. Concurrent Multi-Player Parity Games. In *AAMAS'16*, pages 689–697, 2016.
- 23 F. Mogavero, A. Murano, and L. Sorrentino. On Promptness in Parity Games. In *LPAR'13*, LNCS 8312, pages 601–618. Springer, 2013.
- 24 F. Mogavero, A. Murano, and L. Sorrentino. On Promptness in Parity Games. *Fundam. Inform.*, 139(3):277–305, 2015. doi:10.3233/FI-2015-1235.
- 25 A. Pnueli. The Temporal Logic of Programs. In *FOCS'77*, pages 46–57. IEEE Computer Society, 1977.
- 26 J. P. Queille and J. Sifakis. Specification and Verification of Concurrent Programs in Cesar. In *SP'81*, LNCS 137, pages 337–351. Springer, 1981.