The Descriptive Complexity of Parity Games

Anuj Dawar[1] and Erich Grädel[2]

[1] University of Cambridge Computer Laboratory, Cambridge, CB3 0FD, UK
anuj.dawar@cl.cam.ac.uk
[2] Mathematische Grundlagen der Informatik, RWTH Aachen University, Germany
graedel@logic.rwth-aachen.de

Abstract. We study the logical definablity of the winning regions of parity games. For games with a bounded number of priorities, it is well-known that the winning regions are definable in the modal μ-calculus. Here we investigate the case of an unbounded number of priorities, both for finite game graphs and for arbitrary ones. In the general case, winning regions are definable in guarded second-order logic (GSO), but not in least-fixed point logic (LFP). On finite game graphs, winning regions are LFP-definable if, and only if, they are computable in polynomial time, and this result extends to any class of finite games that is closed under taking bisimulation quotients.

1 Introduction

The question whether the winning regions (i.e. the set of positions from which a particular player has a winning strategy) in parity games can be computed efficiently is one of the most important open problems in the field of infinite games. It is equivalent to the model checking problem for the modal μ-calculus. It is known that the problem is in NP $\cap$ Co-NP, which is a simple consequence of the fact that parity games admit memoryless winning strategies. Much effort has also been put into identifying and classifying special cases of parity games which admit tractable solutions. For instance, there are deterministic polynomial-time algorithms for any class of parity games with a bounded number of priorities, and for parity games with certain restrictions on the underlying game graph, such as bounded tree width, bounded DAG-width, and others.

In this paper we investigate the *descriptive complexity* of parity games, i.e. we ask by what logical means the winning regions of parity games are definable. The descriptive complexity of a problem provides an insight into the structure of the problem, and the sources of algorithmic difficulty, as the logical resources needed to specify the problem are closely tied to its structure. In the case of parity games, the questions that naturally arise are whether the problem is definable in least fixed-point logic (LFP) and in monadic second-order logic (MSO), as these are logics with which it is closely associated. Again, the problem can be considered as solved when we just consider games with a bounded number of priorities. In that case, the winning regions are definable in the modal μ-calculus, and therefore also in monadic second-order logic MSO and in least fixed-point logic

M. Kaminski and S. Martini (Eds.): CSL 2008, LNCS 5213, pp. 353–367, 2008.

LFP. However, the descriptive complexity of parity games with an unbounded number of priorities has not been settled in a satisfactory way. While it can be shown that in this case winning regions are no longer definable in the μ-calculus, the question arises whether they are definable in LFP and/or MSO.

It turns out that the descriptive complexity of parity games on finite game graphs is different from that on arbitrary ones. By making use of the strictness of the alternation hierarchy of LFP on arithmetic, together with an interpretation argument for model checking games, we are able to show that in general, winning regions of parity games are not LFP-definable. On finite game graphs, however, this may well be different. Indeed we prove that the winning regions are LFP-definable if, and only if, they are computable in polynomial-time (despite the fact that, on unordered finite structures, LFP is weaker than Ptime). Our arguments are based on bisimulation quotients of the relational structures that represent parity games and on Otto's result that the bisimulation-invariant fragment of Ptime can be captured by a multi-dimensional variant of the μ-calculus. Our analysis in fact shows that winning regions are LFP-definable on any class of finite parity games for which we have polynomial-time algorithms on their bisimulation quotients. As a consequence, we obtain LFP-definitions for the winning regions of parity games with bounded entanglement. We also show that winning regions are definable, on the class of finite game graphs in stronger fixed-point logics such as NFP and PFP.

A significant open question that remains unanswered is whether winning regions of parity games are MSO-definable. We have not been able to settle this question either on finite game graphs or on arbitrary ones. However, it is not difficult to prove definability in guarded second-order logic (GSO, which allows quantification over sets of edges in addition to sets of vertices) and in Δ_2^1 (and even Δ_1^1 on finite graphs).

We assume that the reader is familiar with the most important logics used in descriptive complexity theory and verification, such as monadic second-order logic MSO, the modal μ-calculus L_μ and the least fixed-point logic LFP. However, we will survey in Section 2 the facts about the logics that we will use.

2 Background from Logic

We assume that the reader is familiar with first-order logic (FO), second-order logic (SO) and monadic second-order logic (MSO), the extension of first-order logic by second-order quantification $\exists X$ and $\forall X$ over *sets* of elements of the structure, on which the formula is evaluated. In contrast to SO where quantification over arbitrary relations (or functions) is admitted, MSO is a much more manageable formalism; in particular it is decidable on many interesting classes of structures (on trees, in particular) and amenable to automata-based methods. We also refer below to fragments of SO defined by bounded quantifer-alternation. In particular, Σ_i^1 consists of those formulae of SO in prenex-normal form with i alternations of second-order quantifiers, so that the first quantifier is existential. Π_i^1 is defined analogously, for formulae starting with a universal quantifier. Finally, Δ^1_i (which is not defined syntactically) is the class of those properties that are definabable by both a Σ^1_i and a Π^1_i formula.

Guarded second-order logic. An interesting extension of MSO is *guarded second-order logic* (GSO), which admits second-order quantification on relations of any arity, provided that they are guarded. Informally, a relation is guarded on a structure $\mathfrak{A}$ if it only consists of tuples whose elements co-exist in some atomic fact of $\mathfrak{A}$. On graphs (V, E), possibly with colourings of nodes and edges, the guarded relations are precisely the relations $R \subseteq V^k$ that only contain tuples $(a_1, \dots, a_k)$, for which either $a_1 = a_2 = \cdots = a_k$, or $\{a_1, \dots, a_k\} = \{u, v\}$ for some edge $(u, v) \in E$. It is not difficult to see that GSO on (coloured) graphs is equivalent to the variant of monadic second-order logic that admits second-order quantification not only over sets of nodes, but also over sets of edges. Further, guarded second-order logic, on any class of structures, is equivalent to monadic second-order logic on the associated incidence graphs [13].

An example showing that GSO is indeed more expressive than MSO, is Hamiltonicity. It is well-known (see e.g. [10]) that Hamiltonicity is not MSO-definable, but it is GSO-definable by the formula

$$\begin{aligned}(\exists H \subseteq E)\big(\forall x\, (\exists^{=1} y\, Hxy \ \wedge\ \exists^{=1} y\, Hyx) \wedge \\ \forall X\, \big[\big(\exists x\, Xx \wedge \forall x \forall y (Hxy \wedge Xx \rightarrow Xy)\big) \rightarrow \forall x\, Xx\big]\big)\end{aligned}$$

which, evaluated on a graph $G = (V, E)$ says that there exists an $H \subseteq E$ with unique successors and predecessors such that (V, H) is connected. This means that G has a Hamilton cycle.

Least fixed point logic. Least fixed-point logic, denoted LFP, extends first order logic by least and greatest fixed points of definable relational operators. We will briefly recall some basic definitions here. For a more extensive introduction to LFP, we refer to [12].

Every formula $\psi(R, \overline{x})$, where R is a relation symbol of arity k and $\overline{x}$ is a tuple of k variables, defines, for any structure $\mathfrak{A}$ of appropriate vocabulary, an update operator $F : \mathcal{P}(A^k) \rightarrow \mathcal{P}(A^k)$ on the class of k-ary relations over the universe A of $\mathfrak{A}$, namely $F : R \mapsto \{\overline{a} : (\mathfrak{A}, R) \models \psi(R, \overline{a})\}$. If ψ is positive in R, that is, if every occurrence of R falls under an even number of negations, this operator is monotone in the sense that $R \subseteq R'$ implies $F(R) \subseteq F(R')$. It is well known that every monotone operator has a least fixed point and a greatest fixed point, which can be defined as the intersection and union, respectively, of all fixed points, but which can also be constructed by transfinite induction.

LFP is defined by adding to the syntax of first order logic the following *fixed point formation rule:* If $\psi(R, \overline{x})$ is a formula with a relational variable R occurring only positively and a tuple of first-order variables $\overline{x}$, and if $\overline{t}$ is a tuple of terms (such that the lengths of $\overline{x}$ and $\overline{t}$ match the arity of R), then $[\mathbf{lfp}\, R\overline{x} \,.\, \psi](\overline{t})$ and $\mathbf{gfp}\, R\overline{x} \,.\, \psi](\overline{t})$ are also formulae, binding the occurrences of the variables R and $\overline{x}$ in ψ.

The semantics of least fixed-point formulae in a structure $\mathfrak{A}$, providing interpretations for all free variables in the formula, is the following: $\mathfrak{A} \models [\mathbf{lfp}\, R\overline{x} \,.\, \psi](\overline{t})$

if $\overline{t}^{\mathfrak{A}}$ belongs to the least fixed point of the update operator defined by ψ on $\mathfrak{A}$. Similarly for greatest fixed points.

Note that in formulae $[\mathbf{lfp}\, R\overline{x}\,.\,\psi](\overline{t})$ one may allow ψ to have other free variables besides $\overline{x}$.

The duality between least and greatest fixed point implies that for any ψ,

$$[\mathbf{gfp}\, R\overline{x}\,.\,\psi](\overline{t}) \equiv \neg[\mathbf{lfp}\, R\overline{x}\,.\,\neg\psi[R/\neg R]](\overline{t}).$$

Using this duality together with de Morgan's laws, every LFP-formula can be brought into *negation normal form*, where negation applies to atoms only.

Other fixed-point logics. We briefly consider two other fixed-point logics that have been studied in the context of finite model theory: PFP, the logic of partial fixed points and NFP, the logic of nondeterministic fixed points. We omit a detailed account of their syntax and semantics here and refer the interested reader to [9]. Here we just note the following facts that will be useful. In terms of expressive power on finite structures, PFP subsumes NFP, which in turn subsumes LFP. On finite *ordered* structures, PFP captures exactly the properties decidable in Pspace and NFP captures the polynomial hierarchy, just as LFP captures Ptime. In the absence of order, it is known that the logics are strictly weaker than the corresponding complexity classes. However, it is also known that the question of the separation of the expressive power of these logics is still equivalent to the separation of the corresponding complexity classes (see [2,1]).

3 Parity Games

Basic definitions. A parity game is specified by a directed graph $G = (V, E)$, with a partition $V = V_0 \cup V_1$ of the nodes into positions of Player 0 and positions of Player 1, and a priority function $\Omega : V \to \omega$. In case $(v, w) \in E$ we call w a successor of v and we denote the set of all successors of v by vE. A *play* in $\mathcal{G}$ is an infinite path $v_0 v_1 \dots$ formed by the two players starting from a given initial position v_0. Whenever the current position v_i belongs to V_0, then Player 0 chooses a successor $v_{i+1} \in v_i E$, if $v_i \in V_1$, then $v_{i+1} \in v_i E$ is selected by Player 1. An infinite play $\pi = v_0 v_1 \dots$ is won by Player 0 if the least priority appearing infinitely often in π is even or no priority appears infinitely often (which may only happen if the range of Ω is infinite). A finite play $\pi = v_0 v_1 \dots v_n$ is won by Player 0 iff $v_n \in V_1$.

A *(deterministic) strategy* for Player σ is a partial function $f : V^* V_\sigma \to V$ that assigns to finite paths through $\mathcal{G}$ ending in a position $v \in V_\sigma$ a successor $w \in vE$. A play $v_0 v_1 \dots \in V^\omega$ is *consistent* with f if, for each initial segment $v_0 \dots v_i$ with $v_i \in V_\sigma$, we have that $v_{i+1} = f(v_0 \dots v_i)$. We say that such a strategy f is winning from position v_0 if every play that starts at v_0 and that is consistent with f is won by Player σ. The *winning region* of Player σ, denoted W_σ, is the set of positions from which Player σ has a winning strategy. A game $\mathcal{G}$ is *determined* if $W_0 \cup W_1 = V$, i.e., if from each position one of the two players has a winning strategy.

Positional determinacy and complexity. Winning strategies can be rather complicated. However, for certain games, including parity games, it suffices to consider *positional strategies*, which are strategies that depend only on the current position, not on the history of the play. A game is *positionally determined*, if it is determined, and each player has a positional winning strategy on his winning region.

It has been proved independently in [11] and [19] that parity games with a finite game graph are positionally determined. The result is extended to infinite game graphs with finitely many priorities in [25]. This has been further extended in [14] to parity games with $\mathrm{rng}(\Omega) = \omega$.

Theorem 1. *Every parity game is positionally determined.*

In a parity game $\mathcal{G} = (V, V_0, V_1, E, \Omega)$, a positional strategy for Player σ, defined on $W \subseteq V$, can be represented by a subgraph $H = (W, S) \subseteq (V, E)$ such that there is precisely one outgoing S-edge from each node $v \in V_\sigma \cap W$ and $vS = vE$ for each node $v \in V_{1-\sigma} \cap W$. On a finite game graph, such a strategy is winning on W if, and only if, the least priority on every cycle in (W, S) has the same parity as σ.

Hence, given a finite parity game $\mathcal{G}$ and a positional strategy (W, S) it can be decided in polynomial time, whether the strategy is winning on W. To decide winning regions we can therefore just guess winning strategies, and verify them in polynomial time.

Corollary 2. *Winning regions of parity games (on finite game graphs) can be decided in* NP $\cap$ Co-NP.

In fact, Jurdziński [16] proved that the problem is in UP $\cap$ Co-UP, where UP denotes the class of NP-problems with unique witnesses. The best known deterministic algorithm has complexity $n^{O(\sqrt{n})}$) [18]. For parity games with a number d of priorities Jurdziński's progress measure lifting algorithm [17] computes winning regions in time $O(dm \cdot (2n/d)^{d/2})$, where m is the number of edges, giving a polynomial-time algorithm when d is bounded.

Parity games as relational structures. To represent parity games as relational structures, so as to make them amenable to a study of logical definability, we consider two different conventions.

For fixed d, we consider the class of parity games with $\mathrm{rng}(\Omega) \subseteq \{0, \dots, d-1\}$, to be given as structures

$$(V, V_0, V_1, E, P_0, \dots, P_{d-1})$$

where E is a binary relation and $V_0, V_1, P_0, \dots, P_{d-1}$ are all unary, with $V = V_0 \cup V_1 = \bigcup_i P_i$, V_0 disjoint from V_1 and the P_i pairwise disjoint. Note that this class of structures (denoted $\mathcal{PG}_d$) is axiomatisable in first-order logic.

On the other hand, to consider the class of parity games with an unbounded number of priorities, we consider them as structures

$$(V, V_0, V_1, E, \prec, \mathrm{Odd})$$

where E, V_0 and V_1 are as before, where $u \prec v$ means that u has a smaller priority than v, and Odd is the set of nodes with an odd priority. We denote this class of structures by $\mathcal{PG}$.

Lemma 3. *The class $\mathcal{PG}$ is axiomatisable in* LFP*, but not in first-order logic.*

Proof. To verify that a structure $\mathcal{G} = (V, V_0, V_1, E, \prec, \mathrm{Odd})$ is a parity game, we have to check that

(1) V is the disjoint union of V_0 and V_1,
(2) $\prec$ is a pre-order on V,
(3) Odd is a union of equivalence classes with respect to the equivalence relation $\approx$ associated with $\prec$ (with $u \approx v$ if $u \not\prec v$ and $v \not\prec u$),
(4) the linear order induced by $\prec$ on the equivalence classes has some order type $\alpha \leq \omega$.

Clearly, (1) – (3) are first-order properties, and it is well-known that order types $\alpha \leq \omega$ can be distinguished from other order types in LFP, but not in first-order logic. □

In each case, when we say that winning regions are definable in a logic L, we mean that there is a formula φ of L such that for any structure $G \in \mathcal{PG}$ (resp. $\mathcal{PG}_d$), φ is true in exactly those nodes in G from which Player 0 has a winning strategy.

Parity Games and LFP. Consider a structure $\mathfrak{A}$ and an LFP-sentence ψ which we may assume to be in negation normal form, without parameters, and *well-named*, in the sense that every fixed-point variable is bound only once. The model checking game $\mathcal{G}(\mathfrak{A}, \psi)$ is a parity game whose positions are formulae $\varphi(\overline{a})$ such that $\varphi(\overline{x})$ is a subformula of ψ, and $\overline{a}$ is a tuple of elements of $\mathfrak{A}$, interpreting the free variables of φ. The initial position is ψ.

Player 0 (Verifier) moves at positions associated to disjunctions and to formulae starting with an existential quantifier. From a position $\varphi \vee \vartheta$ she moves to either φ or ϑ and from a position $\exists y\, \varphi(\overline{a}, y)$ she can move to any position $\varphi(\overline{a}, b)$ for $b \in A$. In addition, Verifier is supposed to move at atomic false positions, i.e., at positions φ of form $a = a'$, $a \neq a'$, $R\overline{a}$, or $\neg R\overline{a}$ (where R is *not* a fixed-point variable) such that $\mathfrak{A} \models \neg\varphi$. However, positions associated with these atoms do not have successors, so Verifier loses at atomic false positions. Dually, Player 1 (Falsifier) moves at conjunctions and universal quantifications, and loses at atomic true positions. In addition, there are positions associated with fixed-point formulae $[\mathbf{fp}\, T\overline{x}\,.\, \varphi(T, \overline{x})](\overline{a})$ and with fixed-points atoms $T\overline{x}$, for fixed-point variables T. At these positions there is a unique move (by Falsifier, say) to the formula defining the fixed point. For a more formal definition, recall that as ψ is well-named, for any fixed-point variable T in ψ there is a unique subformula $[\mathbf{fp}\, T\overline{x}\,.\, \varphi(T, \overline{x})](\overline{a})$. From position $[\mathbf{fp}\, T\overline{x}\,.\, \varphi(T, \overline{x})](\overline{a})$ Falsifier moves to $\varphi(T, \overline{a})$, and from $T\overline{b}$ he moves to $\varphi(T, \overline{b})$.

The priority labelling assigns even priorities to **gfp**-atoms and odd priorities to **lfp**-atoms. Further, if T, T' are fixed-point variables of different kind with

T' depending on T (which means that T occurs free in the formula defining T'), then T-atoms get lower priority than T'-atoms. All remaining positions, not associated with fixed-point variables, receive the least important priority (i.e. the one with the highest value) As a result, the number of priorities in the model checking games equals the alternation depth of the fixed-point formula plus one. For more details and explanations, and for the proof that the construction is correct, see e.g. [12,24].

Theorem 4. *Let ψ be an* LFP*-sentence and $\mathfrak{A}$ a relational structure. Then, $\mathfrak{A} \models \psi$ if, and only if, Player 0 has a winning strategy for the parity game $\mathcal{G}(\mathfrak{A}, \psi)$.*

Moreover, for every structure $\mathfrak{A}$ with at least two elements, and every formula $\varphi(\overline{x}) \in$ LFP the model checking game $\mathcal{G}(\mathfrak{A}, \varphi)$ is first-order interpretable in $\mathfrak{A}$.

Parity games and the modal μ-calculus. It is well-known that there is a very close relationship between parity games and the modal μ-calculus L_μ. First of all, parity games are the model checking games for L_μ. This means that given any formula $\psi \in L_\mu$ and a transition system $\mathcal{K}$, one can build a parity game $\mathcal{G}(\mathcal{K}, \psi)$ which is essentially the product of $\mathcal{K}$ and ψ, such that ψ is true in $\mathcal{K}$ at node v if, and only if, Player 0 has a winning strategy for $\mathcal{G}(\mathcal{K}, \psi)$ starting from the position (v, ψ).

In the other direction, for any fixed d, the winning regions of parity games in $\mathcal{PG}_d$ are definable by a μ-calculus formula

$$\mathrm{Win}_d = \nu X_0 \mu X_1 \nu X_2 \ldots \lambda X_{d-1} \bigvee_{j=0}^{d-1} \big((V_0 \wedge P_j \wedge \Diamond X_j) \vee (V_1 \wedge P_j \wedge \Box X_j) \big).$$

In this formula, the fixed-point operators alternate between ν and μ, and hence $\lambda = \nu$ if d is odd, and $\lambda = \mu$ if d is even.

As a consequence, an efficient algorithm for solving parity games would also solve the model-checking problem for L_μ.

Theorem 5. *For every $d \in \mathbb{N}$, the formula* Win_d *defines the winning region of Player 0 in parity games with priorities* $0, \ldots, d-1$.

In general, formulae of L_μ are hard to read, especially if they have many alternations between least and greatest fixed points, but here we have an elegant argument based on model checking games. Indeed, given any parity game $\mathcal{G} \in \mathcal{PG}_d$, with initial position v, let $\mathcal{G}^*$ be the model-checking game for the formula Win_d on $\mathcal{G}$ so that Player 0 has a winning strategy for $\mathcal{G}^*$ if, and only if, $\mathcal{G}, v \models \mathrm{Win}_d$. It turns out that the game $\mathcal{G}^*$ is essentially the same as the original game $\mathcal{G}$. More precisely, $\mathcal{G}^*$ reduces to $\mathcal{G}$ if we eliminate, for each player, certain moves that would give the opponent the opportunity to win immediately, and if we contract successive moves of the same player into a single move. In particular, Player 0 wins $\mathcal{G}$ if and only if she wins $\mathcal{G}^*$, which is the case if, and only if, the formula Win_d is true at $\mathcal{G}, v$. We refer to [12, Section 3.3.6] for details.

The formulae Win_d also play an important role in the study of the alternation hierarchy of the modal μ-calculus. Clearly, Win_d has alternation depth d and it has been shown that this can not be avoided. As a consequence the alternation hierarchy of the μ-calculus is strict [7,3]. We will need a slightly stronger formulation of this result, for parity games on finite and strongly connected graphs. This easily follows from the general result by the finite model property of the μ-calculus and by a straightforward reduction to strongly connected games.

Theorem 6. *Winning regions in parity games in $\mathcal{PG}_d$ are not definable by formula in the μ-calculus with alternation depth $< d$, even under the assumption that the game graphs are finite and strongly connected.*

4 Definability of Parity Games on Arbitrary Game Graphs

4.1 Definability in Fragments of Second-Order Logic

Which fragments of second-order logic winning are powerful enough to define the winning regions of parity games? In particular, are winning regions definable in monadic second-order logic? While we cannot answer this question in general, we show that it is a direct consequence of positional determinacy that winning regions are GSO-definable. By means of results due to Courcelle, this implies MSO-definability on a number of interesting classes of parity games.

Theorem 7. *Winning regions in $\mathcal{PG}$ are definable in* GSO*, and also in Δ^1_2*

Proof. The same formula can be used to demonstrate both results. Let $z \equiv p$ stand for $\neg(z \prec p) \wedge \neg(p \prec z)$ which says that z and p have the same priority Given a binary relation S, we let

$$\varphi(S,w,p) := \mathbf{gfp}\, Xx\, .\, \mathbf{lfp}\, Yy\, .\, \mathbf{gfp}\, Zz\, .\, \begin{bmatrix} (z \prec p \wedge \forall u(Szu \to Xu)) \vee \\ (z \equiv p \wedge \forall u(Szu \to Yu)) \vee \\ (p \prec z \wedge \forall u(Szu \to Zu)) \end{bmatrix} (w)(x)(y)$$

We claim that this formula defines the set of pairs (w,p) such that every infinite S-path from w either contains infinitely many nodes z with $z \prec p$, or only finitely many nodes with $z \equiv p$.

To see this, consider the model checking game for this formula, played on a graph $(V,S,\prec)$ with fixed nodes w and p. Let $\alpha(z)$ be the subformula inside the fixed point definitions. At position $\alpha(z)$ in the model checking game, Player 0 chooses the right disjunct, according to the priority of z, and then Player 1 takes the game from z to an S-successor u of z; via Xu, Yu or Zu the play then proceeds to position $\alpha(u)$.

Thus a play in the model checking game essentially amounts to the choice of an S-path from w by Player 1. Such a play is won by Player 0 if the outermost fixed point variable seen infinitely often is either X or Z. But this is the case if either a priority $y \prec p$ is seen infinitely often or, if this is not the case, priority p occurs only finitely often on the path.

Thus, the formula

$$\psi(S, w) := \forall p(\mathrm{Odd}(p) \rightarrow \varphi(S, w, p))$$

defines those elements w such that on every infinite S-path from w the lowest priority occurring infinitely often is even. We are now ready to write a formula that defines winning regions of Player 0 for games in $\mathcal{PG}$:

$$(\exists S \subseteq E)[\forall u(V_0 u \rightarrow \exists v Suv) \wedge \forall u \forall v(V_1 u \wedge Euv \rightarrow Suv) \wedge \psi(S, w)]$$

This asserts the existence of a set S of edges which includes at least one outgoing edge for every vertex in V_0 and all the outgoing edges for every vertex in V_1 and such that for this set S, w satisfies ψ. In other words, this formula is true at a vertex w just in case Player 0 has a positional winning strategy from w. Since we know that Player 0 has a positional winning strategy whenever she has any strategy, this is sufficient.

It is easily seen that this formula is in GSO as the binary second-order quantifier on the outside is explicitly guarded and the three occurrences of a fixed-point operator are all monadic and could be therefore replaced by monadic quantifiers.

To see that this also yields a Δ^1_2 definition, we note that any formula of LFP (and hence, in particular, ψ) can be written both as a Σ^1_2 and a Π^1_2 formula (see [9] for a proof). Thus, replacing ψ with its Σ^1_2 equivalent we obtain a Σ^1_2 definition of the winning regions for Player 0. By symmetry, we can also define the winning regions for Player 1 in Σ^1_2 and thereby obtain a Π^1_2 definition for the winning regions for Player 0. □

Although GSO is more expressive than MSO in general, there are a number of classes of graphs where the two logics are equivalent. This collapse occurs in particular over graphs of bounded degree, graphs of bounded tree-width, planar graphs and graphs with an excluded minor. A general result covering and generalising all these graph classes has been proved in [8].

Theorem 8 (Courcelle). GSO *collapses to* MSO *on every class* $\mathcal{C}$ *of graphs that is closed under taking subgraphs and contains only* k*-sparse graphs, for some* $k \in \mathbb{N}$*, (i.e.* $|E| \leq k|V|$ *for every graph* $G = (V, E) \in \mathcal{C}$*).*

The GSO formula we construct in the proof of Theorem 7 contains monadic quantifiers and one second-order quantifier that is guarded by the relation E. In particular, the order $\prec$ does not appear as a guard. From the proof of Courcelle's theorem above, it then follows that this formula is equivalent to an MSO formula on any class of parity games satisfying the sparsity condition above with respect to the number of edges of the game. On each such class, winning regions of parity games are therefore MSO-definable. However, in general, there seems to be no way to eliminate the quantification over a set of edges and we conjecture that this winning regions are not MSO definable. It may be noted that it is possible to reduce parity games on general graphs to games on graphs with outdegree 2 (see, for instance, [17]). While this reduction can be carried out in polynomial time, it involves a quadratic blow-up in the number of vertices and it seems unlikely that it preserves MSO definability.

4.2 Non-definability in Least Fixed-Point Logic

We now show that winning regions of parity games are not definable in LFP when the game graph may be infinite.

Theorem 9. *Winning regions in $\mathcal{PG}$ are not definable in* LFP*, even under the assumptions that the game graph is countable and the number of priorities is finite.*

Proof. Suppose that $\mathrm{Win}(x) \in \mathrm{LFP}$ defines the winning region of Player 0 on $\mathcal{PG}$. We use this formula to solve the model checking problem for LFP on $\mathfrak{N} = (\omega, +, \cdot)$. Recall that, for any $\varphi(x) \in \mathrm{LFP}$, we have a parity game $\mathcal{G}(\mathfrak{N}, \varphi)$ such that, for all n

$$\mathfrak{N} \models \varphi(n) \quad \Longleftrightarrow \quad \mathcal{G}(\mathfrak{N}, \varphi) \models \mathrm{Win}(v_n)$$

(where v_n is the initial position associated with $\varphi(n)$)

Further, the model checking game $\mathcal{G}(\mathfrak{N}, \varphi)$ is first-order interpretable in $\mathfrak{N}$. Hence the formula $\mathrm{Win}(x)$ is mapped, via a first-order translation $\mathfrak{I}_\varphi$, into another LFP-formula $\mathrm{Win}_\varphi(x)$ such that

$$\mathcal{G}(\mathfrak{N}, \varphi) \models \mathrm{Win}(v_n) \quad \Longleftrightarrow \quad \mathfrak{N} \models \mathrm{Win}_\varphi(n).$$

Note that the first-order translation $\mathrm{Win}(x) \mapsto \mathrm{Win}_\varphi(x)$ depends on φ, but does not increase the alternation depth. Hence, on arithmetic, every formula $\varphi(x)$ would be equivalent to one of fixed alternation depth:

$$\mathfrak{N} \models \varphi(n) \quad \Longleftrightarrow \mathfrak{N} \models \mathrm{Win}_\varphi(n).$$

However, it is known that the alternation hierarchy of LFP on arithmetic is strict. □

5 Definability of Parity Games on Finite Graphs

On finite game graphs, the definability issues are different and closely related to complexity. First of all, we observe that winning positions in $\mathcal{PG}$ are definable in Δ^1_1 rather than just Δ^1_2. This is a simple consequence of the fact that the problem of solving parity games is in NP ∩ Co-NP, combined with Fagin's characterisation of NP as the properties that are Σ^1_1 definable.

One of the most interesting questions is whether the winning regions are definable in fixed point logics such as LFP or the μ-calculus. We first observe that the μ-calculus is not sufficient.

5.1 Non-definability in the μ-Calculus

There is a little subtlety involved in considering a μ-calculus on $\mathcal{PG}$, since the priority function is encoded via the pre-order $\prec$. In modal logics, binary relations are handled via modal operators, hence a μ-calculus on $\mathcal{PG}$ would have, for instance, formulae of form $\langle\prec\rangle\varphi$ that hold at a node x just in case φ is true at some node y with lower priority than x (we view $\prec$-edges as going downwards). Further it would be natural to admit also the reverse modality for $\succ$ and possibly others such as $\preceq, \sim, \succeq$ or the associated successor relations. In any case, the precise definition does not really matter, since no formula in any such μ-calculus can define the winning regions of parity games in $\mathcal{PG}$.

Indeed assume that such a formula ψ exists, and let m be its alternation level. For any fixed number d, we can translate ψ into a formula ψ_d of the usual μ-calculus on structures $\mathcal{PG}_d$ which is equivalent to ψ on (strongly connected) parity games with at most d priorities. To define ψ_d we just replace every subformula of form $\langle\preceq\rangle\varphi$ by

$$\bigvee_{0\leq i<j<d} P_j \wedge \mu X.((P_i \wedge \varphi) \vee \Diamond X)$$

which says, for games in $\mathcal{PG}_d$, that from the current node, there is a reachable node of lower priority at which φ is true. Similar constructions work for other modal operators. Note that this translation increases the alternation level of ψ at most by one.

Now take any strongly connected parity game $\mathcal{G} \in \mathcal{PG}$ with priorities $< d$ and let $\mathcal{G}'$ be its presentation as a structure in $\mathcal{PG}_d$. Obviously $\mathcal{G}, v \models \psi$ if, and only if $\mathcal{G}', v \models \psi_d$. But this means that, for any d, the winning regions of parity games with d priorities, on strongly connected finite graphs, can be expressed by a μ-calculus formula with alternation level $m+1$. This contradicts Theorem 6.

Theorem 10. *Winning regions of parity games in $\mathcal{PG}$ (on finite graphs) are not definable in the modal μ-calculus.*

5.2 Definability in Fixed-Point Logics

We now turn to the least fixed point logic LFP. Clearly, a proof that winning regions of parity games in $\mathcal{PG}$ are LFP-definable would imply that parity games are solvable in polynomial time. We show that also the converse direction holds, despite the fact that LFP is weaker than Ptime. To do so, we will use a result due to Martin Otto [23] that the multi-dimensional μ-calculus, which is a fragment of LFP, captures precisely the *bisimulation-invariant* part of Ptime. See also [12, Section 3.5.3] for an exposition of this result.

Winning positions in parity games are of course invariant under the usual notion of bisimulation (e.g. as structures in $\mathcal{PG}_d$). However, to apply Otto's Theorem for parity games with an unbounded number of priorities, we have to consider bisimulation on structures of the form $\mathcal{G} = (V, V_0, V_1, E, \prec, \mathrm{Odd})$. Let $\tau = \{V_0, V_1, E, \prec, \mathrm{Odd}, v\}$ be the vocabulary of parity games with a starting node, and let $\mathrm{Str}(\tau)$ denote the class of all structures of this vocabulary. If we have two such structures that are indeed parity games, then bisimilarity as τ-structures coincides with the usual notion of bisimilarity in $\mathcal{PG}_d$, for appropriate d. However, not all structures in $\mathrm{Str}(\tau)$ are parity games, and the class of parity games is not closed under bisimulation. An efficient procedure for deciding whether a structure is bisimilar to a parity game is to compute its quotient under bisimulation and check whether it is a parity game.

For a structure $(\mathcal{G}, v) \in \mathrm{Str}(\tau)$ consider the bisimulation relation $a \sim b$ on elements of $\mathcal{G}$ defined with respect to the binary relations E, $\prec$ and $\prec^{-1}$. That is to say $\sim$ is the largest relation satisfying:

- if $a \sim b$ then a and b agree on the unary relations V_0, V_1 and Odd;
- for every $x \in aE$ there is a $y \in bE$ such that $x \sim y$, and conversely;
- for every x with $a \prec x$ there is a y with $b \prec y$ and $x \sim y$ and conversely; and finally
- for every $x \prec a$ there is a $y \prec b$ such that $x \sim y$, and conversely.

We write $(\mathcal{G}, v)^{\sim}$ for the *bisimulation quotient* of $(\mathcal{G}, v)$, i.e. the structure whose elements are the equivalence classes in $\mathcal{G}$ with respect to $\sim$ with the relations $V_0, V_1, E, \prec, \mathrm{Odd}$ defined in the natural way and $[v]$ as the starting vertex.

Lemma 11. *A structure $(\mathcal{G}, v) \in \mathrm{Str}(\tau)$ is bisimilar to a parity game if, and only if, its bisimulation quotient is a parity game, i.e. $(\mathcal{G}, v)^{\sim} \in \mathcal{PG}$.*

Proof. The direction from right to left is obvious. For the other direction, it suffices to establish that the bisimulation quotient of a parity game is itself a parity game, since the quotient of a structure bisimilar to $(\mathcal{G}, v)$ is isomorphic to the quotient of $(\mathcal{G}, v)$. But, it is easily verified that each of the four conditions given in the proof of Lemma 3 is preserved under taking quotients. □

Theorem 12. *Let $\mathcal{C}$ be any class of parity games on finite game graphs, such that winning positions on its bisimulation quotients are decidable in polynomial time. Then, on $\mathcal{C}$, winning positions are* LFP*-definable.*

Proof. Let $\mathrm{Win}\mathcal{C}$ be the class of parity games $(\mathcal{G}, v)$, such that $(\mathcal{G}, v) \in \mathcal{C}$, and Player 0 wins from initial position v. It suffices to construct a bisimulation-invariant class X of structures $(\mathcal{H}, u)$ such that

(1) X is decidable in polynomial time.
(2) $X \cap \mathcal{C} = \mathrm{Win}\mathcal{C}$.

Indeed, by Otto's Theorem X is then definable by an LFP-formula $\psi(x)$, such that, given any parity game $(\mathcal{G}, v) \in \mathcal{C}$ we have

$$\mathcal{G}, v \in \mathrm{Win}\mathcal{C} \iff \mathcal{G}, v \in X \iff \mathcal{G} \models \psi(v).$$

By assumption, there exists a polynomial time algorithm A which, given a parity game $(\mathcal{G}, v) \in \mathcal{C}^{\sim}$, decides whether Player 0 wins $\mathcal{G}$ from v. It is not important what the algorithm returns for quotients outside $\mathcal{C}^{\sim}$, as long as it is isomorphism-invariant and halts in polynomial time. Finally, let B be the algorithm which, given any finite structure in $\mathrm{Str}(\tau)$, first computes its bisimulation quotient, and then applies algorithm A.

Clearly B is a polynomial time algorithm, since bisimulation quotients are efficiently computable. Further the class X of structures accepted by B is invariant under bisimulation. Indeed, let $\mathcal{H}$ and $\mathcal{H}'$ be two bisimilar structures. Then their bisimulation quotients are isomorphic and are therefore either both accepted or both rejected by A. Finally, $X \cap C = \mathrm{Win}\mathcal{C}$. Indeed, given a parity game $\mathcal{G}, v \in \mathcal{C}$, then it has the same winner as its bisimulation quotient which is therefore correctly decided by the algorithm B. □

Corollary 13. *On the class $\mathcal{PG}$ of all finite parity games, winning regions are* LFP*-definable if, and only if, they are computable in polynomial time.*

We can deduce more from this construction. The proof of Otto's theorem relies on an LFP-definable interpretation of an *ordered* bisimulation quotient $\mathcal{G}^{\sim}$ within $\mathcal{G}$. Now, on ordered structures, every problem decidable in PSPACE is definable in PFP and every problem in NP is definable in NFP. Since winning regions on the class $\mathcal{PG}$ are computable in these complexity classes, we can compose the formulas of PFP and NFP with the LFP interpretation to obtain formulas which do not require an order. Thus we have the following theorem.

Theorem 14. *On the class $\mathcal{PG}$ of all finite parity games, winning regions are definable in the logics* PFP *and* NFP.

5.3 Restricted Classes

While the question of whether or not winning regions in finite parity games can be computed in polynomial-time remains open, many restricted classes of games have been investigated on which polynomial-time algorithms for computing winning regions (and winning strategies) are known. These include parity games with a bounded number of priorities [17], games where even and odd cycles do not intersect, solitaire games and nested solitaire games [5], and parity games of bounded tree width [20], bounded entanglement [6], bounded DAG-width [4,21], bounded Kelly-width [15], or bounded clique width [22].

In view of Theorem 12 the question arises which of these classes are closed under taking bisimulation quotients. This is not the case for games of bounded tree width or of bounded clique width, since one can easily construct trees whose bisimulation quotients contain arbitrarily large grids.

However, it is the case for games of bounded entanglement, and possibly also for other classes defined by specific measures for directed graphs.

Entanglement is a parameter for the complexity of finite directed graphs which measures to what extent the cycles of the graph are intertwined. It is defined by means of a game played by a thief against k detectives according to the following rules. Initially the thief selects an arbitrary position v_0 of the given graph $G = (V, E)$ and the detectives are outside of G. In any move the detectives may either stay where they are, or place one of them on the current position v of the thief. The thief tries to escape by moving to a successor $w \in vE$ that is not occupied by a detective. If no such position exists, the thief is caught and the detectives have won. Note that the thief sees the move of the detectives before he decides on his own move, and that he has to leave his current position no matter whether the detectives stay where they are or not. The entanglement of G, denoted $\mathrm{ent}(G)$, is the minimal number $k \in \mathbb{N}$ such that k detectives have a strategy to catch the thief on G.

It has been proved in [5] that, for any fixed k, winning positions in parity games with entanglement at most k are computable in polynomial time. Our results complement this by showing that on such classes, winning positions are LFP-definable. By Theorem 12 it suffices to show that the entanglement cannot increase when we move from a directed graph G to its bisimulation quotient $G^{\sim}$.

Lemma 15. *For any directed graph $\mathcal{G}$, $\mathrm{ent}(G^{\sim}) \leq \mathrm{ent}(G)$.*

Proof. We show that whenever the thief has a strategy to escape against k detectives on $G^{\sim}$, he also has a similar strategy on G, by which he stays clear of not only the nodes occupied by the detectives, but also all nodes bisimilar to these. Suppose that the detectives occupy nodes $w_1, \dots, w_k$ of G and the thief has to move from his current position u, For the corresponding situation on $G^{\sim}$ the thief can move according to his escaping strategy from the bisimulation class $[u]$ to a successor $[v] \in [u]E^{\sim}$ that is not occupied by any of the detectives, i.e. $[v] \neq [w_i]$ for all i. Since there is an edge from $[u]$ to $[v]$ in $G^{\sim}$ there exists a node $v' \sim v$ with $(u, v') \in E$. In G, the thief moves to any such node $v' \in [v] \cap uE$. Since $[v'] = [v] \neq [w_i]$ for all i, the thief maintains the property that he avoids in G all positions bisimilar to those occupied by a detective. □

Corollary 16. *In any class of parity games of bounded entanglement, the winning positions are definable in* LFP.

6 Conclusions

While the exact computational complexity of computing winning regions in finite parity games remains a much studied open question, we have addressed a related question that has not received much attention—that of the descriptive complexity of parity games, both on finite and arbitrary structures. While we are able to settle the definability questions in many interesting cases, a significant open question that remains is whether winning regions are definable in MSO. Interestingly, the question of whether winning regions are definable in LFP on finite game graphs turns out to be equivalent to the long-standing open question of whether they are computable in Ptime.

References

1. Abiteboul, S., Vardi, M.Y., Vianu, V.: Fixpoint logics, relational machines, and computational complexity. J. ACM 44(1), 30–46 (1997)
2. Abiteboul, S., Vianu, V.: Computing with first-order logic. Journal of Computer and System Sciences 50(2), 309–335 (1995)

3. Arnold, A.: The mu-calculus alternation-depth is strict on binary trees. RAIRO Informatique Théorique et Applications 33, 329–339 (1999)
4. Berwanger, D., Dawar, A., Hunter, P., Kreutzer, S.: Dag-width and parity games. In: Durand, B., Thomas, W. (eds.) STACS 2006. LNCS, vol. 3884, pp. 524–536. Springer, Heidelberg (2006)
5. Berwanger, D., Grädel, E.: Fixed-point logics and solitaire games. Theory of Computing Systems 37, 675–694 (2004)
6. Berwanger, D., Grädel, E.: Entanglement - A measure for the complexity of directed graphs with applications to logic and games. In: Baader, F., Voronkov, A. (eds.) LPAR 2004. LNCS (LNAI), vol. 3452, pp. 209–223. Springer, Heidelberg (2005)
7. Bradfield, J.: The modal μ-calculus alternation hierarchy is strict. Theoretical Computer Science 195, 133–153 (1998)
8. Courcelle, B.: The monadic second-order logic of graphs XIV: Uniformly sparse graphs and edge set quantifications. Theoretical Computer Science 299, 1–36 (2003)
9. Dawar, A., Gurevich, Y.: Fixed point logics. Bulletin of Symbolic Logic 8, 65–88 (2002)
10. Ebbinghaus, H.-D., Flum, J.: Finite Model Theory, 2nd edn. Springer, Heidelberg (1999)
11. Emerson, A., Jutla, C.: Tree automata, mu-calculus and determinacy. In: Proc. 32nd IEEE Symp. on Foundations of Computer Science, pp. 368–377 (1991)
12. Grädel, E., et al.: Finite Model Theory and Its Applications. Springer, Heidelberg (2007)
13. Grädel, E., Hirsch, C., Otto, M.: Back and forth between guarded and modal logics. ACM Transactions on Computational Logic 3, 418–463 (2002)
14. Grädel, E., Walukiewicz, I.: Positional determinacy of games with infinitely many priorities. Logical Methods in Computer Science (2006)
15. Hunter, P.: Complexity and Infinite Games on Finite Graphs. PhD thesis, University of Cambridge (2007)
16. Jurdziński, M.: Deciding the winner in parity games is in UP ∩ Co-UP. Information Processing Letters 68, 119–124 (1998)
17. Jurdziński, M.: Small progress measures for solving parity games. In: Reichel, H., Tison, S. (eds.) STACS 2000. LNCS, vol. 1770, pp. 290–301. Springer, Heidelberg (2000)
18. Jurdziński, M., Paterson, M., Zwick, U.: A deterministic subexponential algorithm for solving parity games. In: Proceedings of ACM-SIAM Proceedings on Discrete Algorithms, SODA 2006, pp. 117–123 (2006)
19. Mostowski, A.: Games with forbidden positions. Technical Report Tech. Report 78, University of Gdansk (1991)
20. Obdrzálek, J.: Fast mu-calculus model checking when tree-width is bounded. In: Hunt Jr., W.A., Somenzi, F. (eds.) CAV 2003. LNCS, vol. 2725, pp. 80–92. Springer, Heidelberg (2003)
21. Obdrzálek, J.: DAG-width - connectivity measure for directed graphs. In: Proceedings of ACM-SIAM Proceedings on Discrete Algorithms, SODA 2006, pp. 814–821 (2006)
22. Obdrzálek, J.: Clique-width and parity games. In: Duparc, J., Henzinger, T.A. (eds.) CSL 2007. LNCS, vol. 4646, pp. 54–68. Springer, Heidelberg (2007)
23. Otto, M.: Bisimulation-invariant Ptime and higher-dimensional mu-calculus. Theoretical Computer Science 224, 237–265 (1999)
24. Stirling, C.: Bisimulation, model checking and other games. Notes for the Mathfit instructional meeting on games and computation. Edinburgh (1997)
25. Zielonka, W.: Infinite games on finitely coloured graphs with applications to automata on infinite trees. Theoretical Computer Science 200, 135–183 (1998)